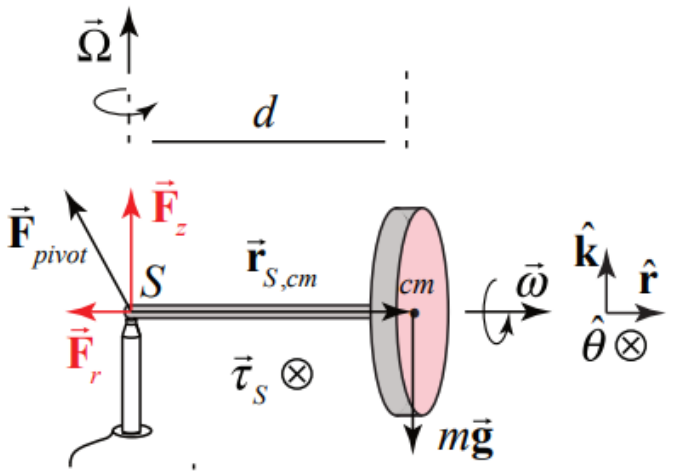
Anti-gravitational motor,

Myth or … Reality?

Part1: Inspiration



When the wheel turns, the torque effect exerted by the arm is eliminated. Only the weight should be supported by Fz

$$Fz - m*g = 0$$

Inertial Wheel ➡ https://www.youtube.com/watch?v=GeyDf4ooPdo

Part 2:

Theory

$$Fgravitational(Newton) = \left(\frac{K_g}{m}\right) = M\ m * \frac{G}{d^2}$$ On earth surface → = 9,8 * m

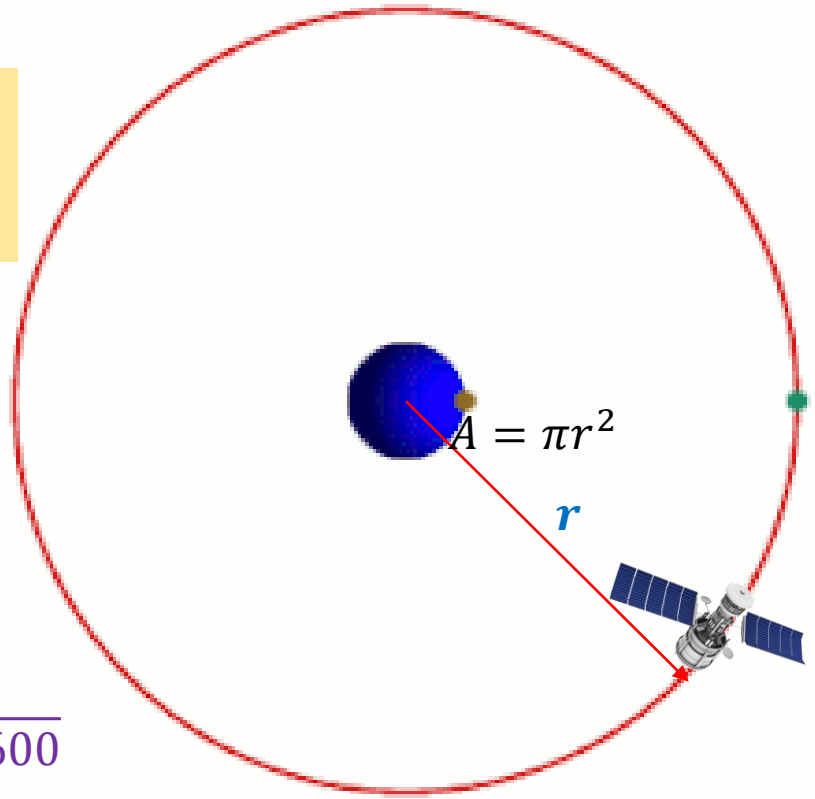
$$Fcentrifugal(Newton) = K_g * \frac{rad}{sg} * metro = \frac{m}{r} v^2 = \mathbf{m * \omega^2 * r}$$

Geostationary orbit of satellite

Pay attention on this case :
m_1 is the weight of satellite but is compensated with m_2

$$m_1 = m_2$$

$$\omega\left(\frac{rad}{sg}\right) = \left(\frac{1\ turn}{day}\right) = \frac{2 * \pi}{24h * 3600}$$



Geoestationary orbit r doesn't depend of satellite mass, only depends of ω

$$r^3 = \frac{G * M_{\text{earth}}}{\omega^2}$$

$r \sim$ 42.000 km

Fg (gravitational) = Fc (centrifugal)

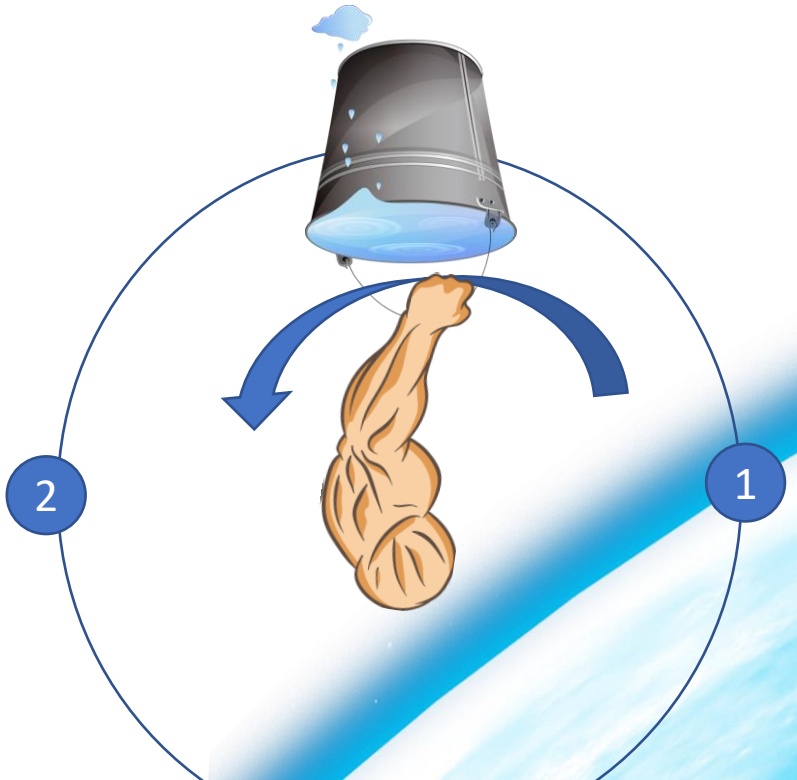
Distance to center of earth

Rotation earth

$$Fgravitational(Newton) = \left(\frac{Kg}{m}\right) = M_{\text{earth}} * \frac{G}{r^2} * m_2 = Fcentrifugal(Newton) = Kg * \frac{rad}{sg} * metro = m_1 * \omega^2 * r$$

But…. Our goal is to get an **antigravity engine** on earth surface

…. Also we have discovered that centrifugal force is an enemy of ***gravitational*** **force**



Question: What speed do I need to avoid that water drops ? if:

- r is our arm (1 meter)
- m_1 is water bucket (1 kg)

Fg= 9.8 Newton = **Fc= 1 * 1* ω^2** ➡ ω =3.13 radianes/sg

1 radian is aprox 57 degrees so $\frac{\omega}{(2\pi)}$ ➡ **180º/sg**

Conclusion : almost from pos 1 to pos 2……. in 1 sg

Part 3:

Antigravitational fundament

Try to imagine a **water bucket** that spin :

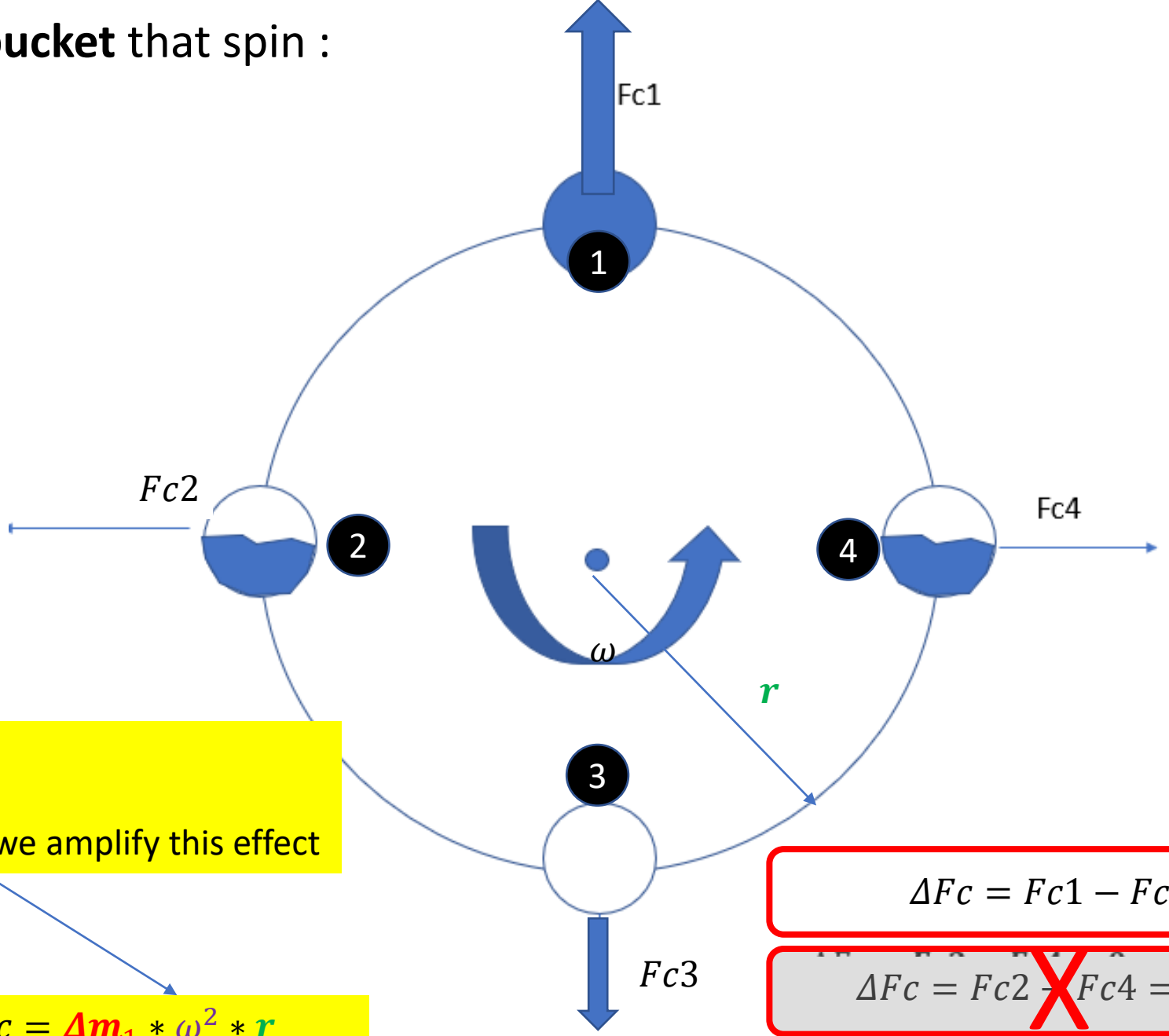
- full of water **on top**
- and empty **on botton**

Fc2 cancels with **Fc4**

The ΔFc **is due** the difference of m_1

We call also estimate that

Δm_1 **= is the weight of water**



Very important:

If we increase ω^2 we amplify this effect

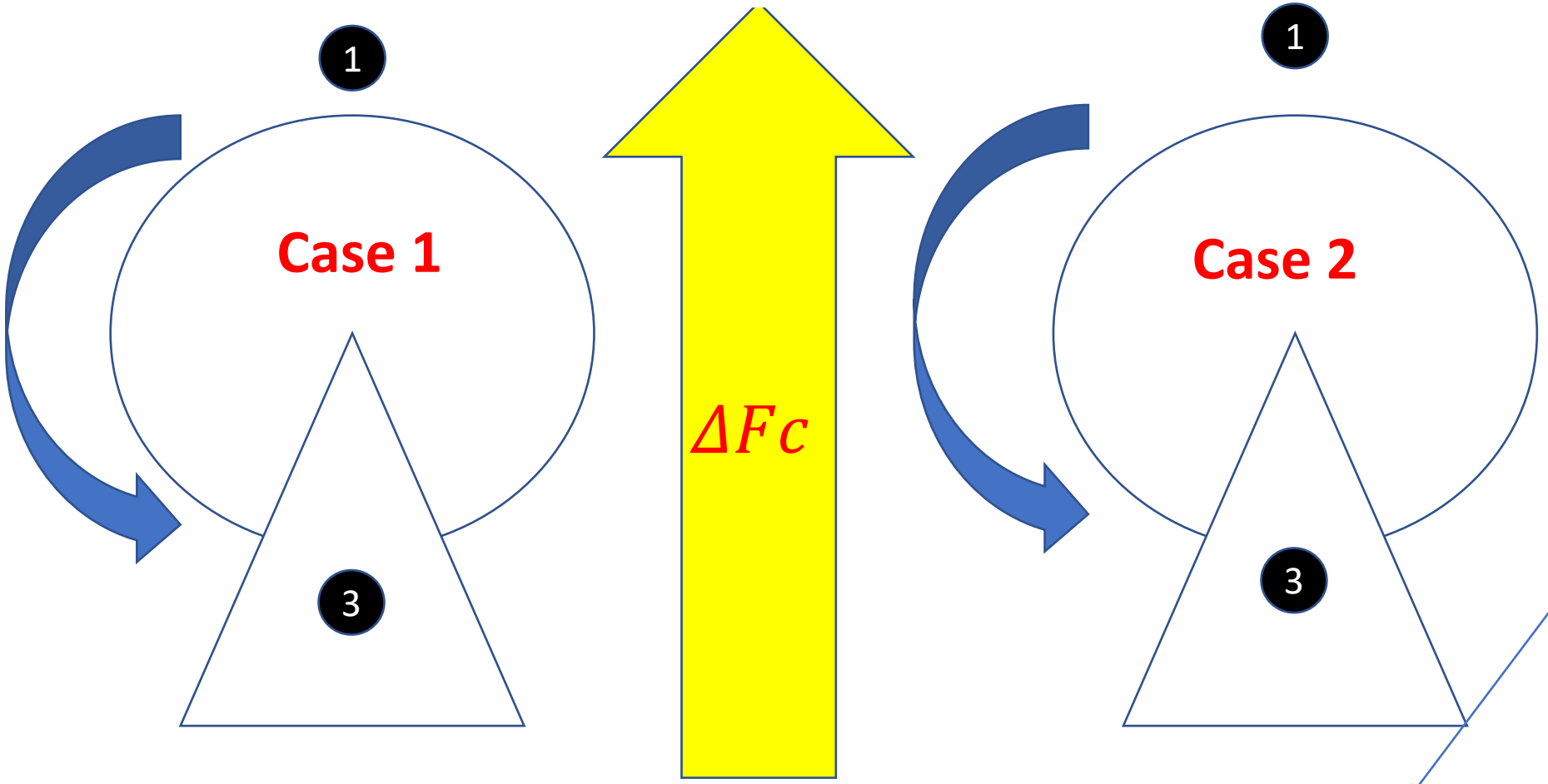
$$\Delta Fc = \Delta m_1 * \omega^2 * r$$

$$\Delta Fc = Fc1 - Fc3$$

$$\Delta Fc = Fc2 - Fc4 = 0$$

let's reflect: we have to focus

$m_2{}^*$ = Note que m2 is the mass of overall system



Please note that

ω^2

Will help us to "amplify" this effect

$9{,}8 * m_2{}^* = \Delta m_1 * \omega^2 * r$

$9{,}8 * m_2{}^* = m_1 * \omega^2 * \Delta r$

Case 1: If our rotating system change the mass on position 1 and 3, **we'll have one antigravitational engine**

Case 2: If our rotating system change the radio on position 1 and 3 **we'll have one antigravitational engine**

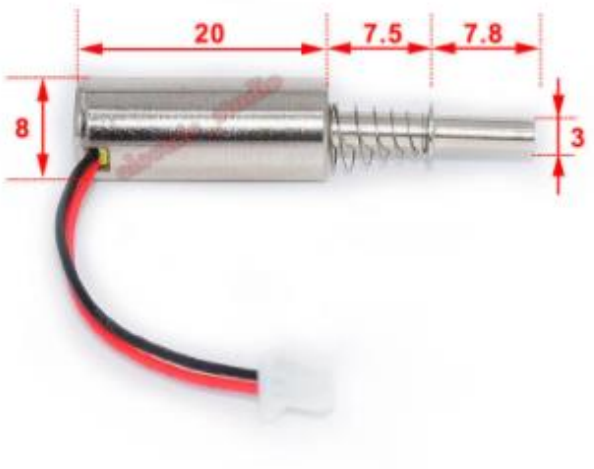
Part4:
Let’s implement

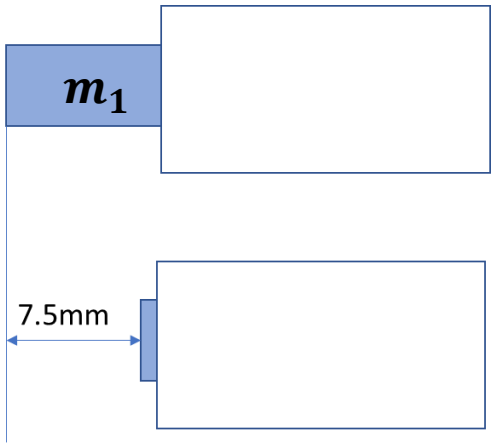
It’s impossible?

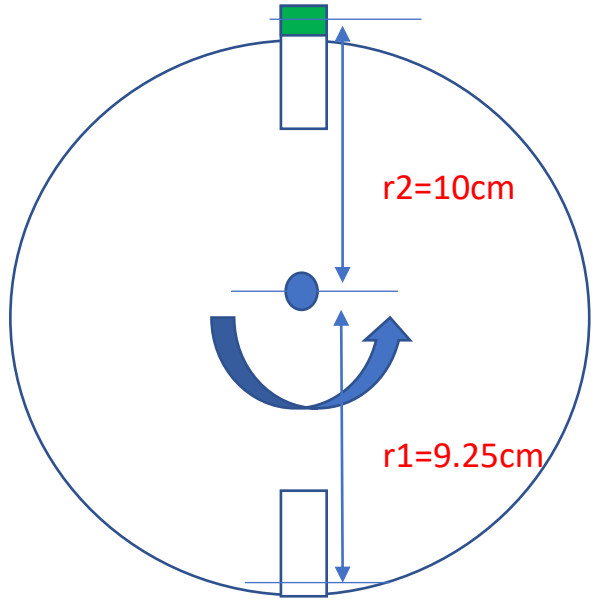
Don’t forget: **all download links are in the last page: designs STL, Arduino code source, Documentation**

Prototype 1: CASE 2=> Δr

Electromagnetic plunger system activated by electric impulse







We need a mass m_1 that change "r" on pos 1 and 3:

m_1=2 gr

$\Delta r = r_2 - r_1 = 7.5mm$

Estimate speed= 1800 rpm= 30 turns by second

Estimate m_2= 400 grams

$$\Delta Fc = 0.002 * \left(\frac{1800}{60} * 2\pi\right)^2 * 0.0075 = 0.53\ N$$

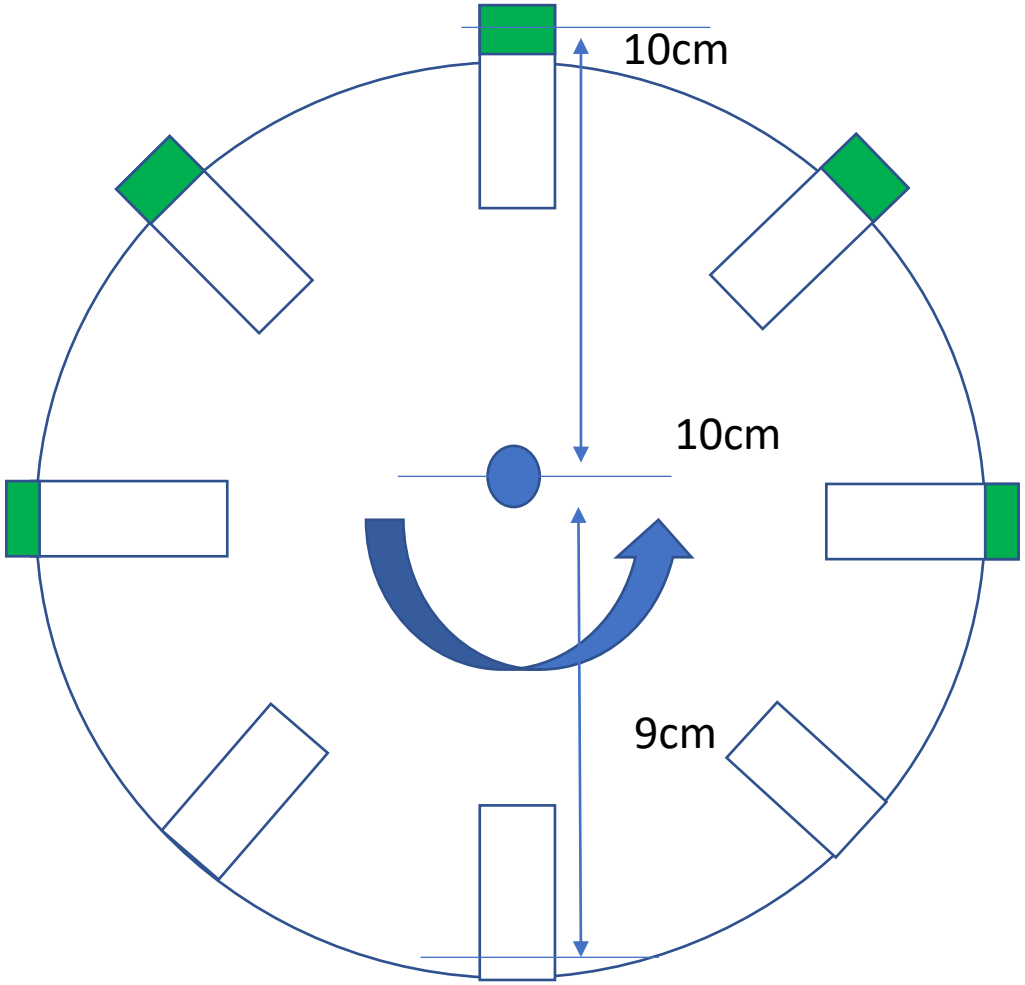
$$Fg = 9.8 * 0.4 = 3.92\ N$$

FAIL

Fg=3.92 > Fc= 0.53 N

Or not?

Prototype 1: Electromagnetic plunger system activated by electric impulse



We can add **N plunger** symmetrically to avoid eccentricities

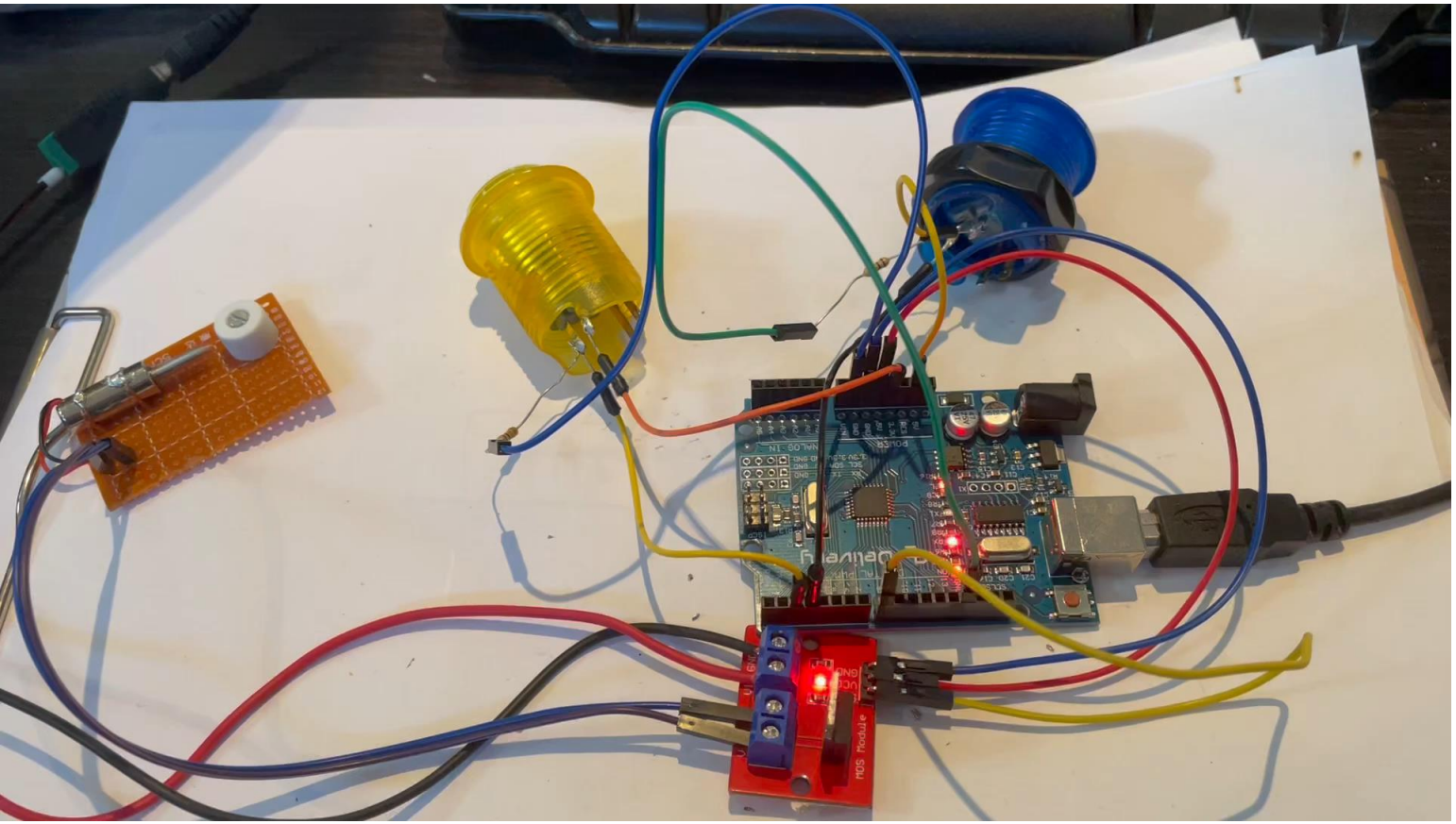
IF N=8 → $8 * \Delta Fc = 4.24\ N > Fg = 3.92N$

…….. but

Prototype 1: **Electromagnetic plunger system activated by electric impulse**

Drawbacks:

The speed of 1800 rpm (30 Hz) is not possible **for a full stroke(7.5mm),** the inertia of piston and the elongation of the spring prevent it, (for example, try operating a subwoofer at 1 kHz).



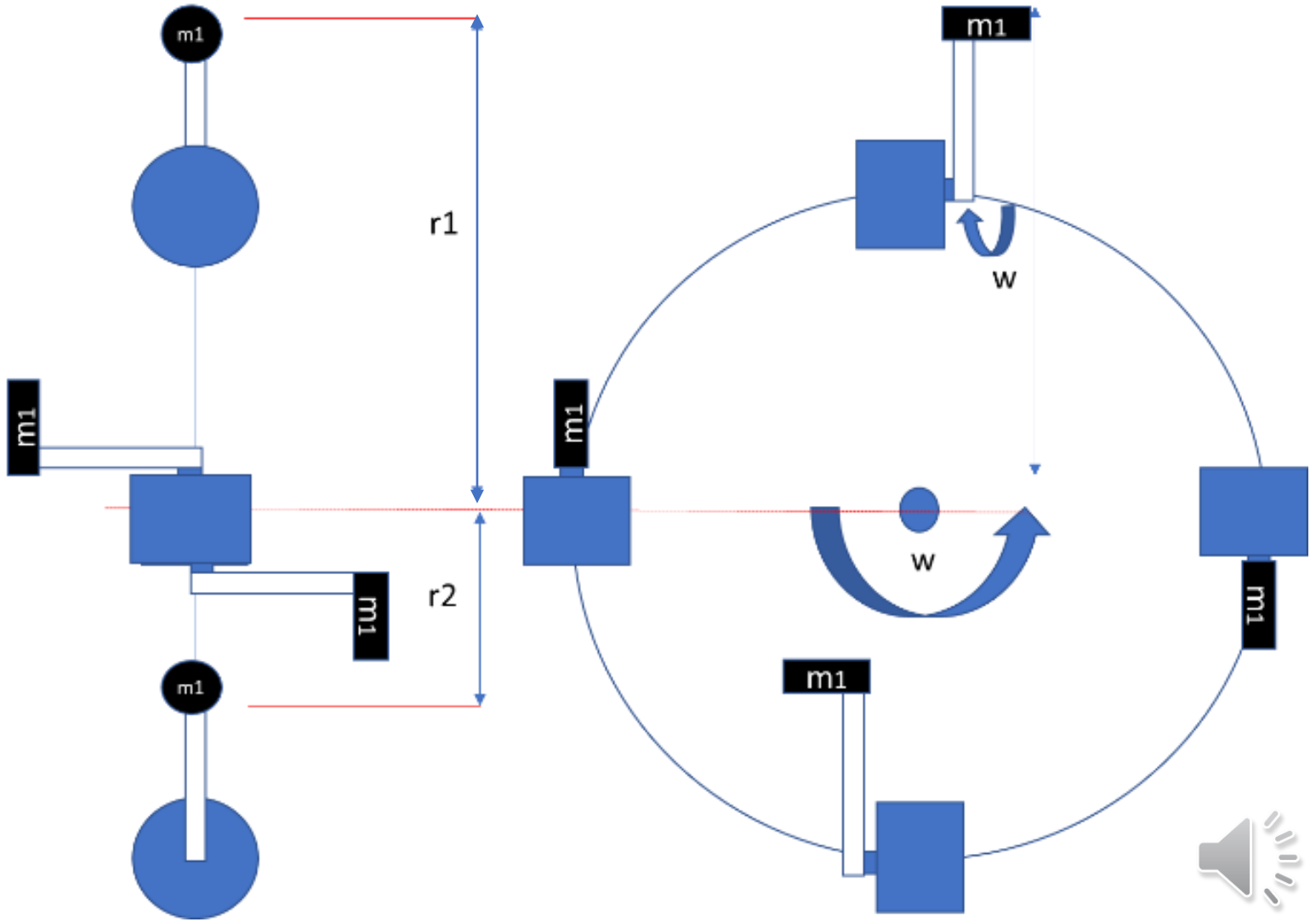
Other problem arises as the plunger **is based on an electromagnet**, the heating of the solenoid and the high energy consumption also invalidate this solution.

Prototype 2: CASE 2=> Δr

System with step motors

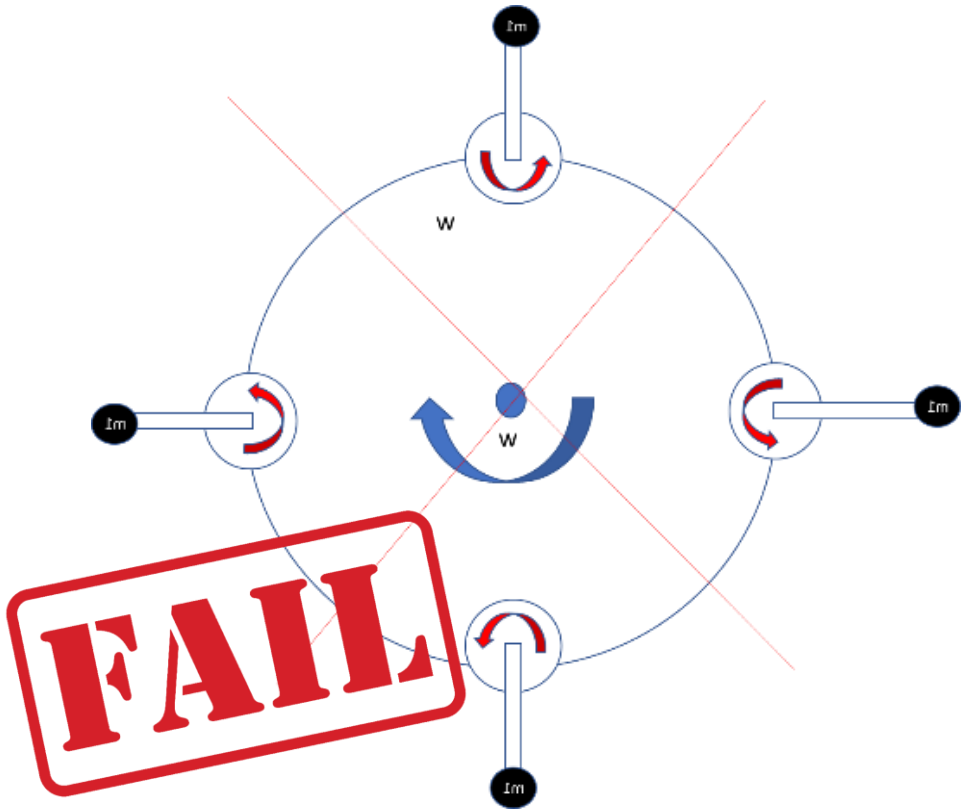
We replace the piston with a **step motor** (we'll call inertial motor)
that rotates synchronously to the traction motor

and **produces the effect of changing the radius** depending on its <u>angular position</u>



Prototype 2: System with step motors

We discard the configuration of **" inercial motor"** rotating **in same plane** that **"traction motor"** because will generate **opposite Fc** between motors



we tested 28BYJ-48 + driver ULN2003

Advantages:
- High torque
- Fast implementation
- lightweight

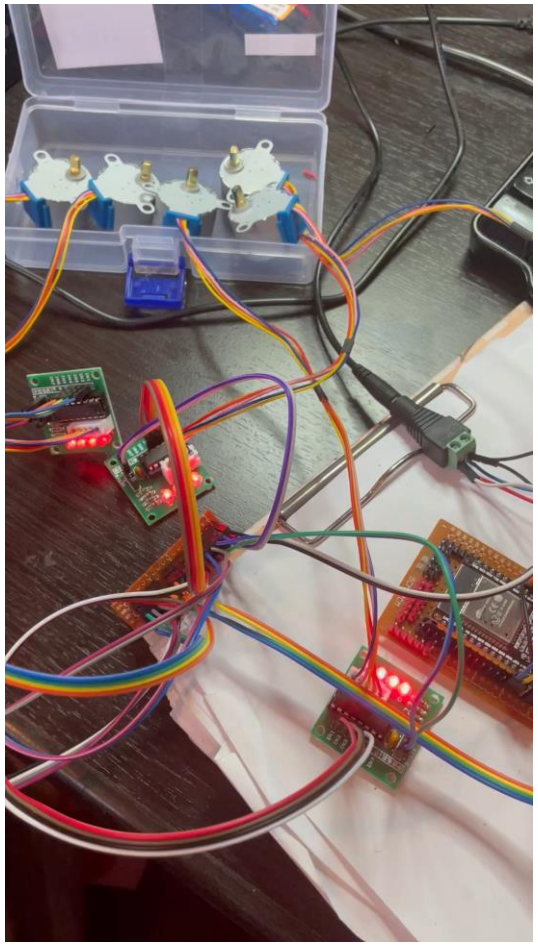
Drawbacks:
- Slow speed

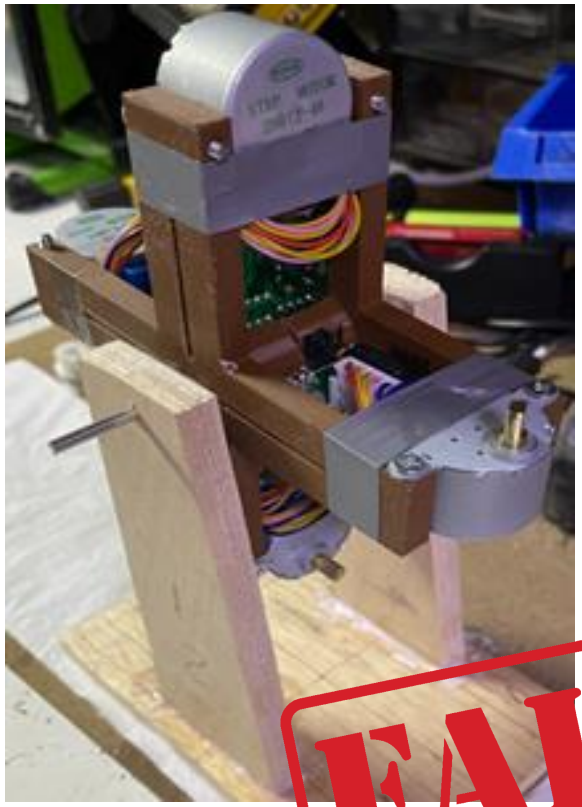
Prototype 2: System with step motors

Testing on sandbox (prototype 2.1):

- We'll use ESP32 as driver processor with Wifi(Rx)
- We'll use ESP32 as control processor with Wifi(Tx)
- 7.5v/3000mAh lithium battery

Oops!!! we forgot "tracking motor"





FAIL

We have got some important advances:
Motor synchronizing and drivers **works fine as slaves**

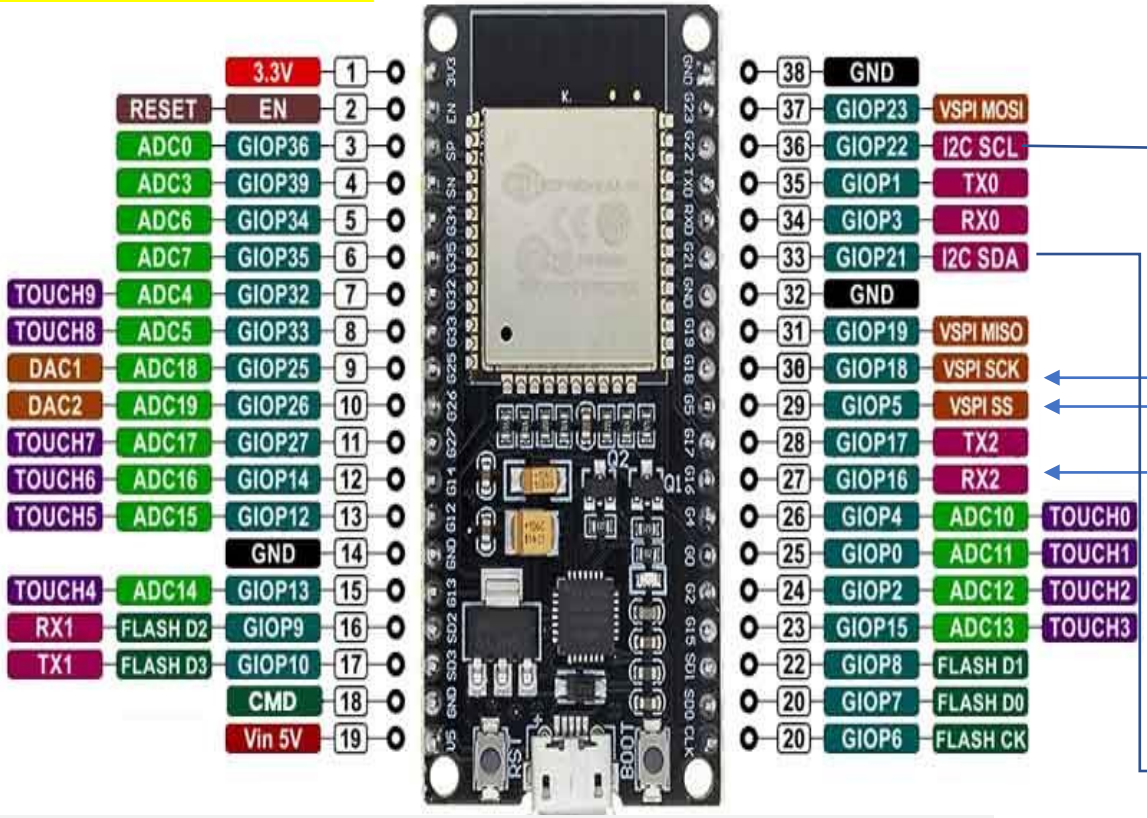
PINOUT

ESP32 38 PINES ESP WROOM 32

Prototype 2: System with step motors

```
#define LED_BUILTIN 2
#define RESET   16
#define BTNDOWN   5
#define BTNUP   18
```

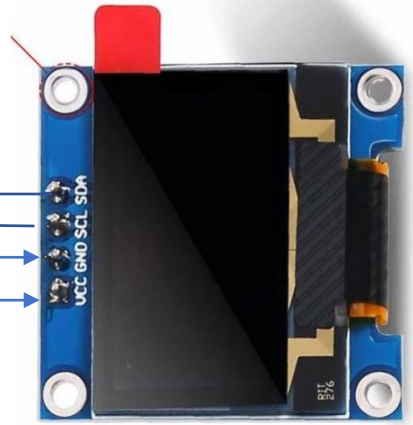
TX Design



3v3

Pullout resistors

MCU AVR Game 5D



Display SSD1306
128x64

GND

3v3

```
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels
#define OLED_RESET     -1
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);
```

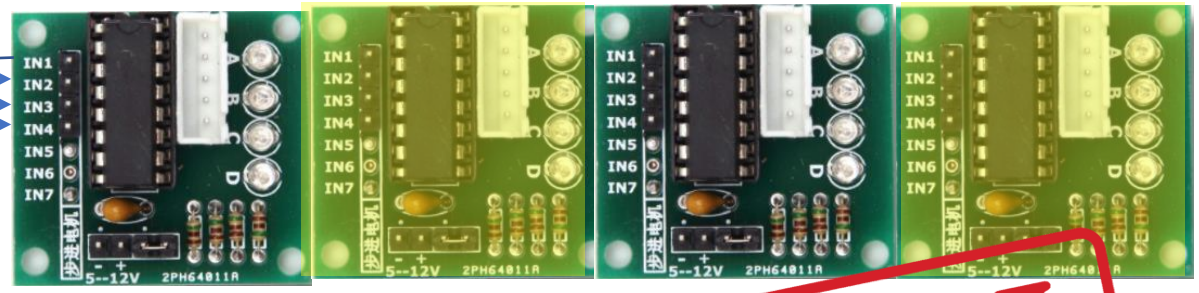
* Source code link in last page

Prototype 2: System with step motors

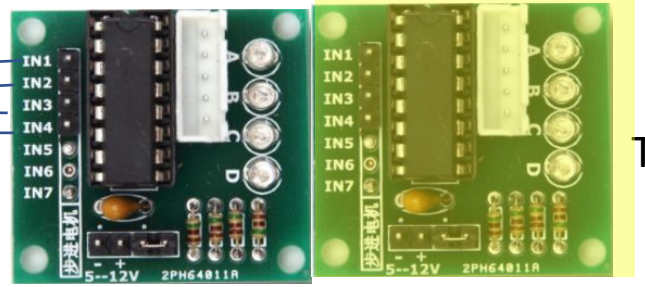
RX Design


```
//----------------------------traccion motor
//clockwise order In1, In2,In3,In4
//counterclockwise  order In2, In1,In4,In3
#define motorPin1  18     // IN1 on the ULN2003 driver
#define motorPin2  19     // IN2 on the ULN2003 driver
#define motorPin3  12     // IN3 on the ULN2003 driver
#define motorPin4  13     // IN4 on the ULN2003 driver
```

```
//----------------------------inercial motor
//clockwise order In1, In2,In3,In4
//counterclockwise  order In2, In1,In4,In3
#define motorPin1  14     // IN1 on the ULN2003 driver
#define motorPin2  15     // IN2 on the ULN2003 driver
#define motorPin3  16     // IN3 on the ULN2003 driver
#define motorPin4  17     // IN4 on the ULN2003 driver
```



Inertial motors



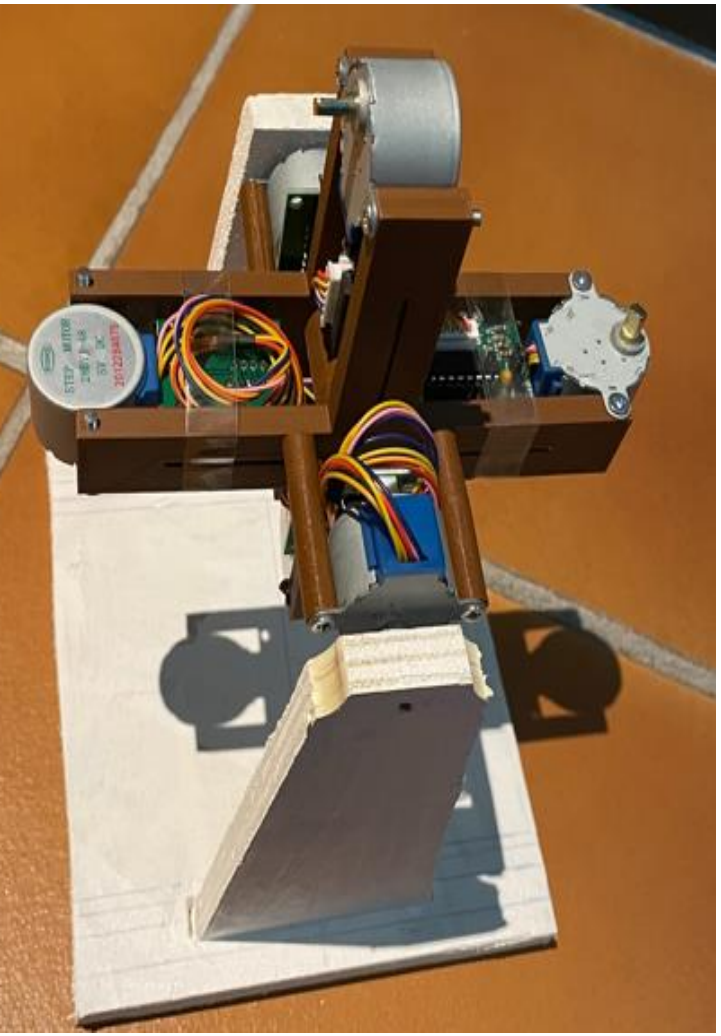
Traction motors

FAIL

* Mote that some drivers require reverse connection

Prototype 2: System with step motors

Testing on sandbox (prototype 2.2):

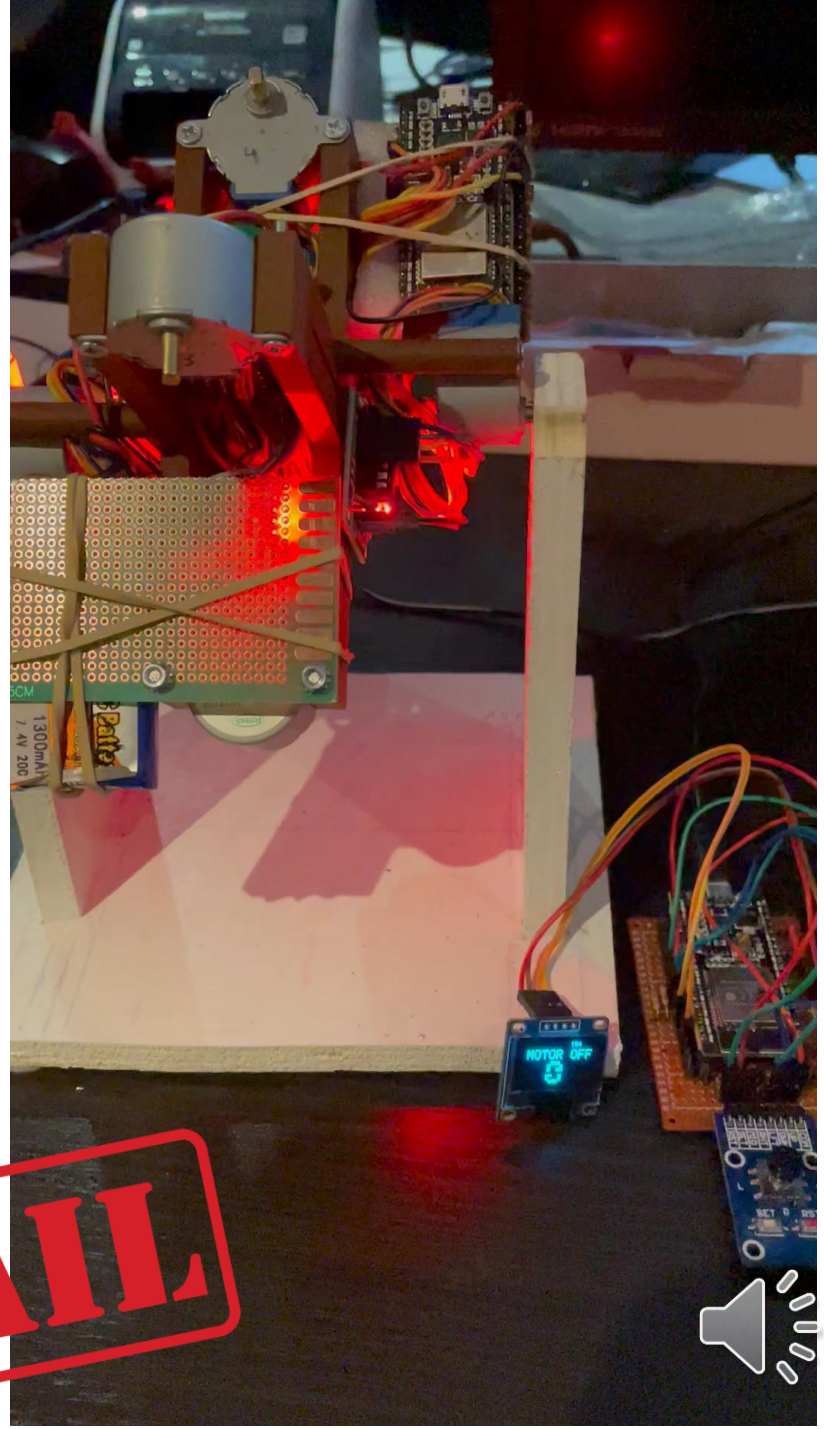


Oops!!! we forgot that : “tracking motor” doesn’t like eccentricity of batteries

We have got some important advances:

software RX needs implementation with Arduino + FreeTOS

- receive commands from Tx
- give movement instructions to the engine, which we remember must be synchronous. Varies by speed, but signal may be required every 100 usg



Prototype 2: System with step motors

Arduino+FreeTOS in RX

```
void setup(){
//--------------------------------------------------- FREE_TOS
xTaskCreatePinnedToCore(
                  Wifi_RX,        // Task function.
                 "DATA_rx",       // String with name of task.
                 5000,          // Stack size in bytes.
                 NULL,           // Parameter passed as input of the task
                 1,             // Priority of the task.
                 NULL,            // Task handle.
                 CONFIG_ARDUINO_RUNNING_CORE);    // important

}
void Wifi_RX(void * parameter){
//---------------------------------FreeTOS process
  for(;;){
}
}

void loop(){
//-------------------------------------------motor
if((power==1) && (up_command > 0)){
   stepper1.setSpeed(speed);
   stepper1.runSpeed();
   stepper2.setSpeed(speed);
   stepper2.runSpeed();
}
}
```

Arduino+FreeTOS in TX

```
void setup(){
//--------------------------------------------------- FREE_TOS
xTaskCreatePinnedToCore(
                 display_job,        /* Task function. */
                 "Intermitente",       /* String with name of task. */
                 5000,          /* Stack size in bytes. */
                 NULL,           /* Parameter passed as input of the task */
                 1,             /* Priority of the task. */
                 NULL,            /* Task handle. */
                 CONFIG_ARDUINO_RUNNING_CORE);    /* important. */
}

void display_job(void * parameter){
//---------------------------------FreeTOS process
  for(;;){
}
}

void loop(){
//-----------------------------Main process
}
```

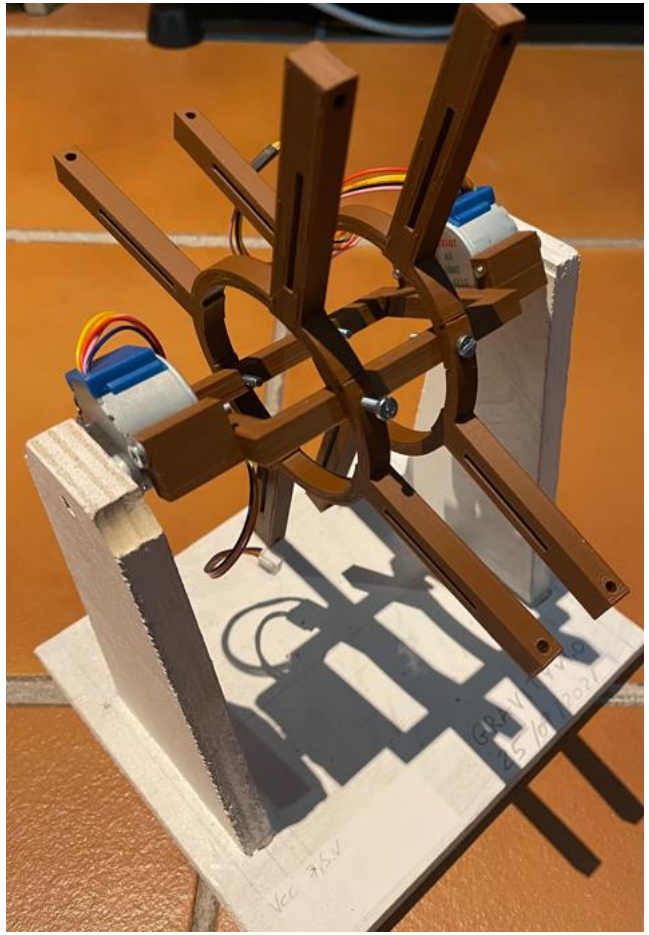
FreeTos provide real time process management

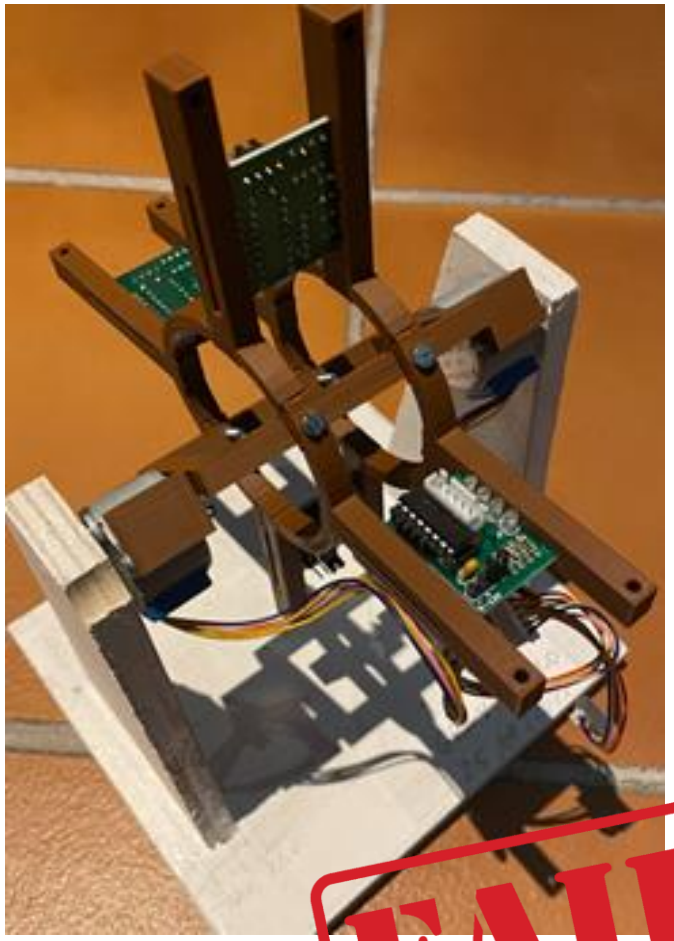
Prototype 2: System with step motors

Testing on sandbox (prototype 2.3):

Oops!!! speed too low:

We knew low speed drawback
But anyway we have got final structure design





w = 17 rpm

$m_1 = 9$ gr

$\Delta r = r_2 - r_1 = 9$ cm

$m_2 = 600$ gr

- battery 200 gr
- 6 motor 300 gr
- wood stand + 3d structure

$$\Delta Fc = 0.009 * \left(\frac{17}{60} * 2\pi\right)^2 * 0.09 = \mathbf{0.0026\ N}$$

$$Fg = 9.8 * 0.6 = \mathbf{5.88\ N}$$

FAIL

$\Delta Fc * 4 = \mathbf{0.0026} * 4\ N$ = 0,0104 N

Prototype 3: CASE 2=> Δr

System with high speed step motors motor YK36BYG12 + driver A4988

- This driver requires **Vref** & **Imax** setting
- Positive pole of **volt_meter** over potentiometer gives **Vref**
- **We have got max speed & torque with Vref 0,5 volt**
- Consider that VDD is 3v3 because is the ESP32 ‘s Vout

Modelo	A4988
Color	Verde o Rojo
Intensidad máxima	2A
Tensión máxima	35V
Microsteps	16
Rs típico	0.05, 0.1 o 0.2
Fórmulas	I_max = Vref / (8 * Rs)
	Vref = I_max * 8 * Rs

- Weight 52 gr
- 4 wire Bipolar
- Step angle 0.9 degrees
- Phase resistance 20 ohm
- Torque ?

No too much heavy

We need new driver **A4988**

Requires 400 steps to get a turn

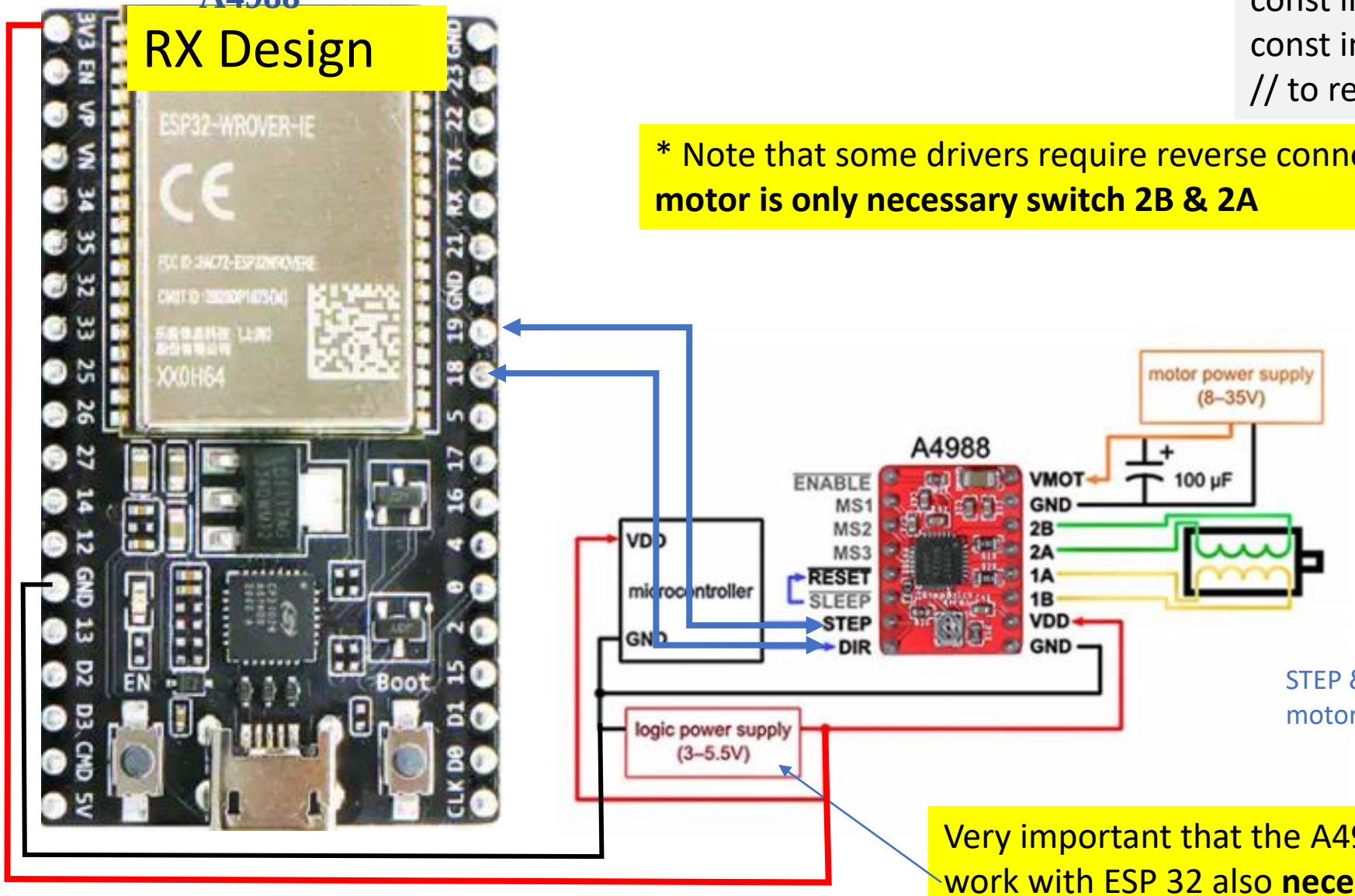
High resistance means low current=> low torque

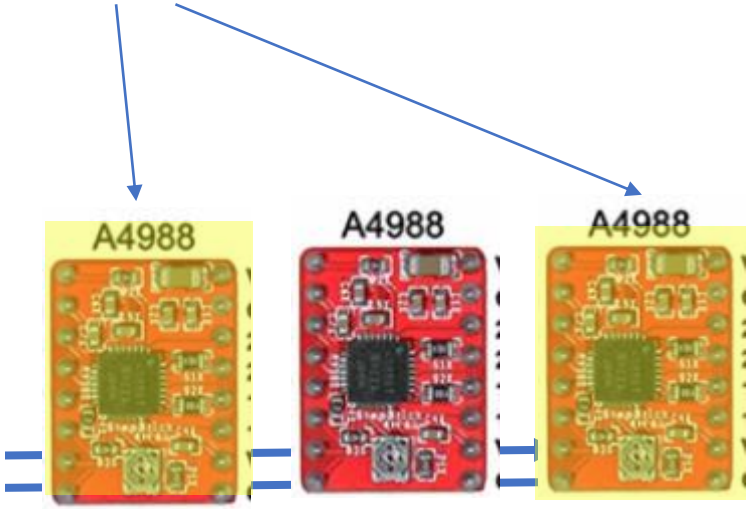
Prototype 3.1: 4 units step motor YK36BYG12 + driver A4988

RX Design

```
const int steps = 400;
const int dirPin = 18;
const int stepPin = 19;
// to reverse motors
```

* Note that some drivers require reverse connection in step motor, **on bipolar motor is only necessary switch 2B & 2A**

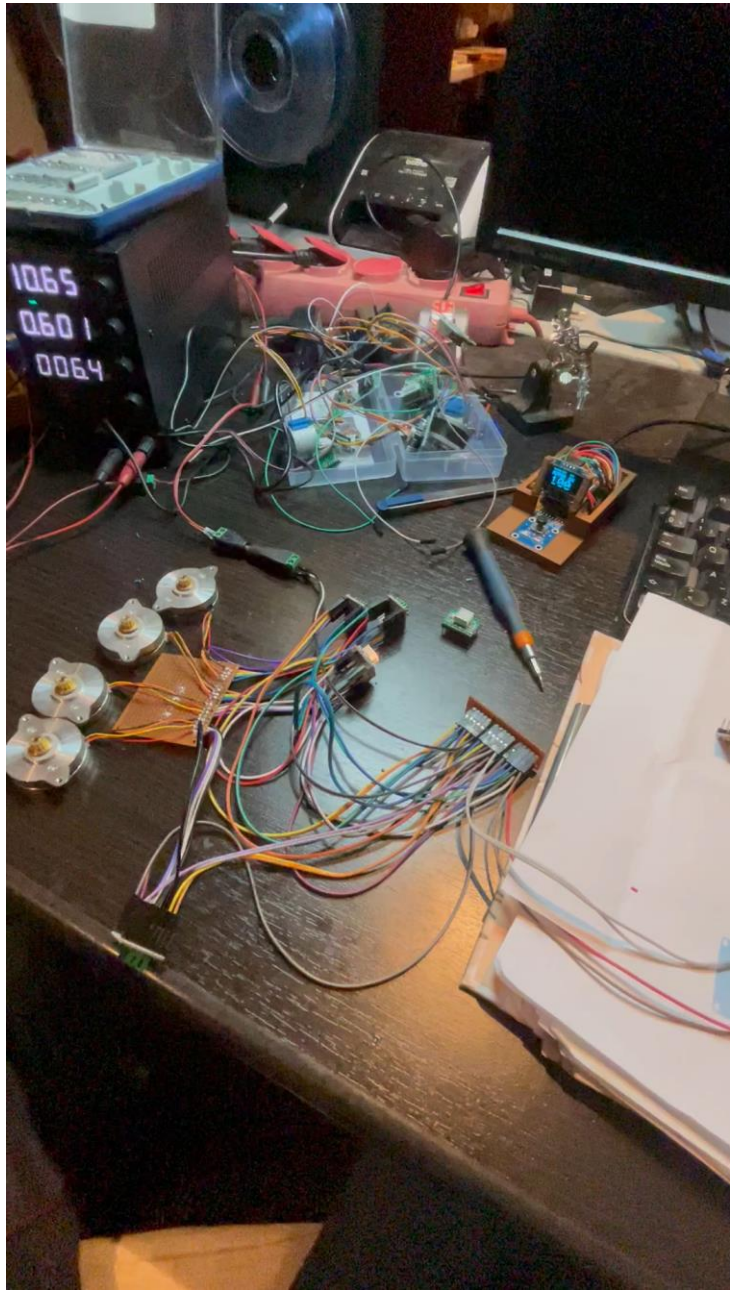




STEP & DIR signals are connected in parallel to get all motor synchronized

Very important that the A4988 **supports digital levels of 3v3** to work with ESP 32 also **necessary for STEP & DIR signals**

Prototype 3.1:4 units step motor YK36BYG12 + driver A4988



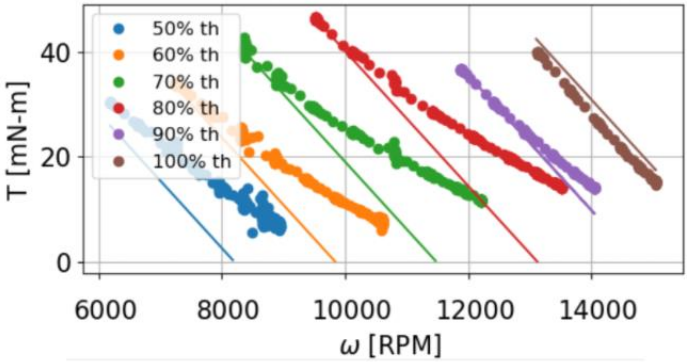
Testing on sandbox (prototype 3.1):

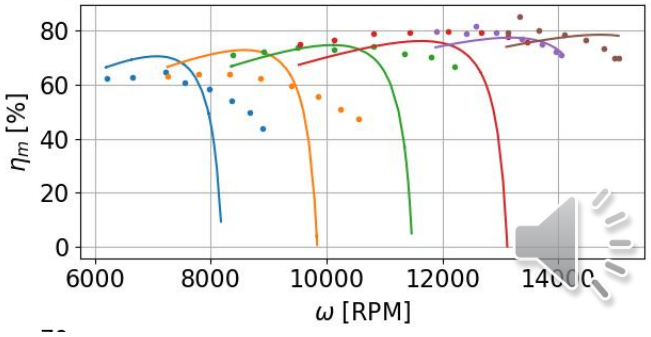
- We have received only 4* motor units (2 traction+2 inertial)
- We'll use ESP32 as driver processor with Wifi(Rx)
- We'll use ESP32 as control processor with Wifi(Tx)
- 7.5v/3000mAh lithium battery

After practical implementation

- Max speed is w=250 rpm with the necessary torque to move a weight of 9 gr on their arm
- m_1=9 gr
- Δr=9 cm,
- m_2=800 gr

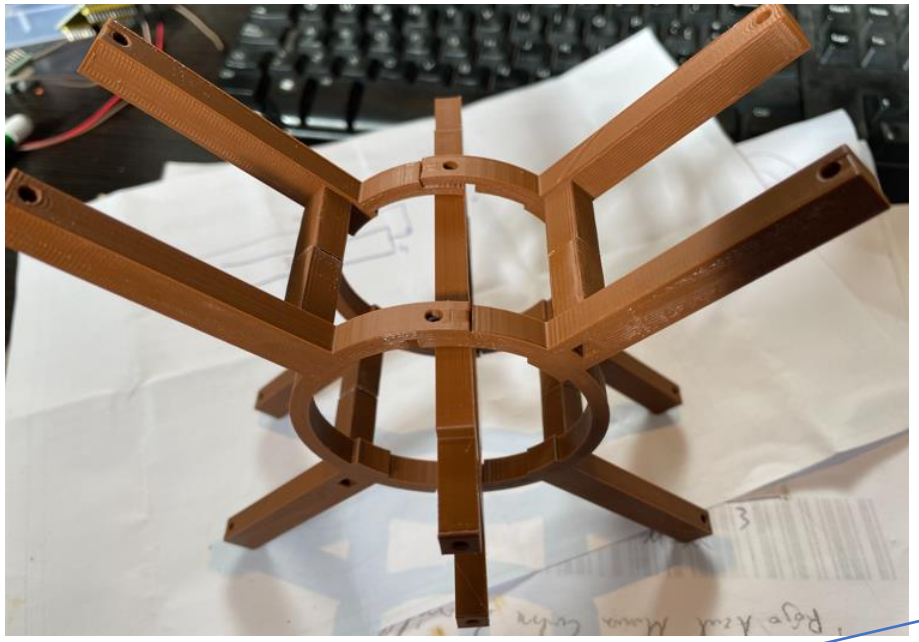
An important drawback, step motor **torque is reduced wen speed is increased**





*stock issue

Prototype 3.1:4 step motor YK36BYG12 + driver A4988



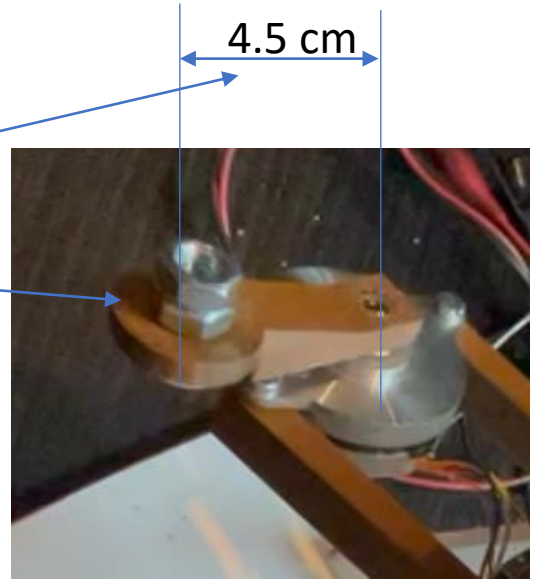
$$\Delta Fc = 0.009 * \left(\frac{250}{60} * 2\pi\right)^2 * 0.09 = 0.55\ N$$

$$Fg = 9.8 * 0.8 = 7.84\ N$$

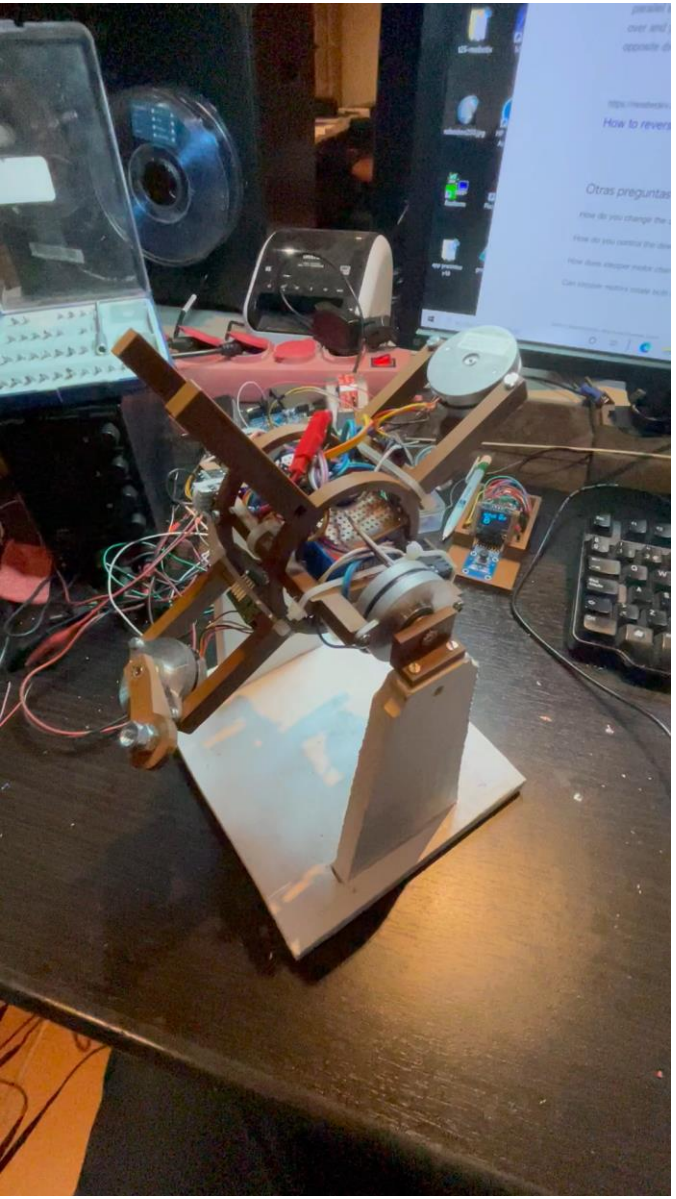
$$2 * \Delta Fc = 2 * 0.55 = 1.10N\ < 7.84\ N$$

FAIL

- m_1=9 gr
- Δr=9 cm
- m_2=800 gr
- ω =250 rpm



Prototype 3.1: 4 step motor YK36BYG12 + driver A4988



… what do we need to get levitation => more speed
=> 2 additional step motor

If we get ω =500 rpm (just double than previous speed)

- m_1=9 gr
- Δr=9 cm
- m_2=800 gr
- ω =250 rpm

$$\boldsymbol{\Delta Fc} = 0.009 * \left(\frac{500}{60} * 2\pi\right)^2 * 0.09 = 2.22\ N$$

$$\boldsymbol{Fg} = 9.8 * 0.8 = 7.84\ N$$

$$\boldsymbol{4 * \Delta Fc} = 4 * 2.22 = \boldsymbol{8.88\ N} > \boldsymbol{7.84\ N}$$

Prototype 4.0

We need :

- your help
- Your ideas
- Your improvements

I hope I have motivated you with this dream

Download links in nex