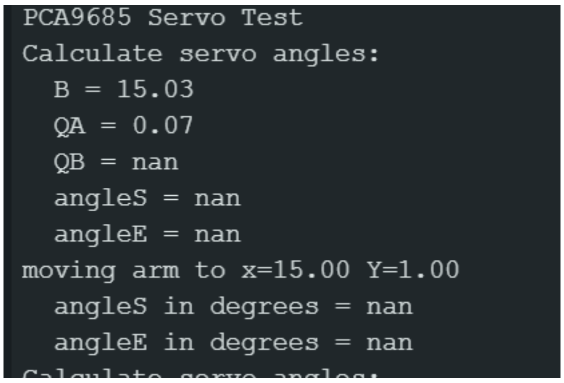
Session 13

Wanting to test my sketch code, the measurements of the angles were not realized they were considered as indefinite numbers (fig 1), so I started the session by looking for the source of the error.



```
PCA9685 Servo Test
Calculate servo angles:
  B = 15.03
  QA = 0.07
  QB = nan
  angleS = nan
  angleE = nan
moving arm to x=15.00 Y=1.00
  angleS in degrees = nan
  angleE in degrees = nan
Calculate servo angles:
```

Fig 1: angles are considered as indefinite numbers.

1. the cosine function is bounded between [0,1], in the case of the movement of my arm we are even deprived of the 0 since the angle of movement of the shoulder is calculated according to :

```
float QB = acos(tvala / tvalb);
```

➔ It was therefore necessary to limit the space of movement.

2. THE PCA9685 servomotor driver only works under pulse, an angle/pulse analogy was then necessary, on the data sheet of the TD-8135 MG 35kg the pulse varies between 5000-2500 uses, with a protractor, we have determined the angle corresponding to the min and max pulse then a pulse angle linearity relation.

```
//analogie angle impulsion
int IMP [valeur] ={2500,2400,2300,2200,2100,2000,1900,1800,1700,1600,1500,1400,1300,1200,1100};
int ANG [valeur] ={0,11,22,33,44,55,66,77,88,99,110,121,132,143,154};
```

➔ The operating range was then limited between 11° and 154°

Then we set up a function transforming into an impulse to be transmitted to the PCA9685

```
int valeur_angle(int angle)
{
  int pulse_wide, analog_value;
  pulse_wide   = map(angle, 0, 180, pulse_min, pulse_max);
  analog_value = int(float(pulse_wide) / 1000000 * frequence * 4096);
  return analog_value;
}
```

This function is certainly functional but not the most precise, it has therefore been replaced by another transforming the angles into microseconds, which involves pca9685.writeMicroseconds.

```
int angle_to_pulse(int angle){
  return 2500 - angle*9 ;
}
```

Finally, we did the same for the servomotors responsible for the movement of the elbow. The **movearm [1]** function is now perfectly usable with two degrees of freedom. we've gone to 3 degrees including the arm/clamp junction now.

[1} featured in report 12