VIII Premi PRBB al millor treball de recerca en Ciències de la Salut i de la Vida

2013

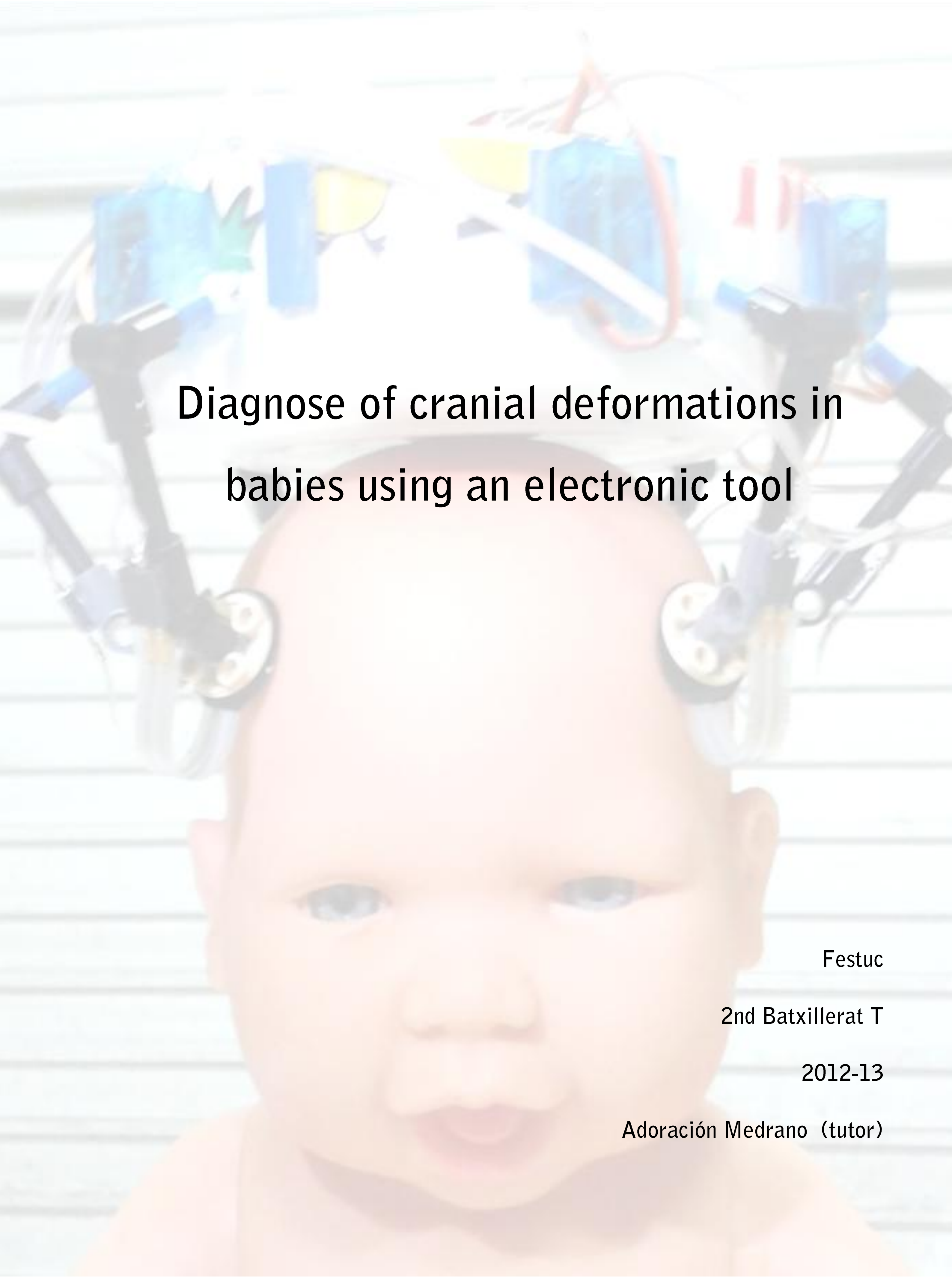
Treball guanyador del 3er premi

Diagnose of cranial deformations in babies using an electronic tool

Jordi Figols Vilarrasa

Tutora: Adoración Medrano

Institut Lliçà (Lliçà de Munt)



Diagnose of cranial deformations in babies using an electronic tool

Festuc

2nd Batxillerat T

2012-13

Adoración Medrano (tutor)

Index

Annexes:

DVD:

1.Presentation

1.1.Introduction

The aim of this research project is to study the cranial deformations in babies and create an electronic tool capable of improving its detection by creating a reliable way to measure the shape of the crane.

Nowadays, checking the shape of newborn's cranes is a regular task in paediatric primary attention, however, there is not an exact and precise way to diagnose specific types of cranial deformations and the diagnosis relies on the ability and perception of the doctor or the nurse.

After seeing this gap in medical methods I want to fix it by providing an electronic device capable of diagnosing deformations precisely.

1.2.Hypothesis

An electronic device is capable of improving the current method of diagnosing cranial deformations in babies.

1.3.Objectives

General Objectives:

- Create a reliable tool able to offer an improvement of the current method.
- Provide a better way to follow the growth of deformations.
- Build a simple and handy electronic device.

Secondary objectives:

- Learn more about programming and robotics.
- Become aware of how the diagnostic system works.
- Learn to work with an electronic platform.
- Become able to do a successful research project.

- Create a practical method that will really be used.

Personal objectives:

- Link the careers of my family (mother nurse, father engineer).
- Do something for the development of the society.
- Support the research of my region.
- Practise my English expression to gain ability at making my ideas understood.

1.4.Motivation

An important reason why I have chosen to do this project is that I will be able to apply one of my interests to the medicine and the development of technology. This has been my most remarkable source of motivation since the thought that I can help somebody by doing an enjoyable task has kept in my mind encouraging me to work harder.

Besides, the fact of having a physical working tool as a result of my research project has also been my incentive, as I believe that, psychologically, seeing your work ended brings a sense of relief and fullness.

Finally, it is worth mentioning that, maybe, I will approach my career to some kind of product development; therefore, this project is useful to me as a training and formation.

2.Theoretical part

2.1Cranial deformations

Cranial deformations in newborns are usual during the first days of life as a consequence of the osseous adaptation when going through the birth canal. After the first six weeks they get solved spontaneously.

Nevertheless, there are cranial deformations which appear or remain and if they are not treated, they can end up in severe aesthetic or health problems.

We must differentiate these deformities that are caused by a dysfunction of the osseous development which tend to be antenatal, craniosynostosis, from the ones that are due to external causes after the birth, branchycephaly and postural plagiocephaly.

The craniosynostosis is the fusion and the early closure (during the antenatal period) of one or some cranial sutures.

During the normal development of the fetus and the baby the different bones that make up the crane must not be united so as to allow the fast growth during the first months. The early closure of the cranial sutures produces malformations that mean aesthetic and mental development troubles as they interfere in brain’s growth.

The most frequent craniosynostosis are the ones that only fusion one suture and they are named depending on the suture they affect (sagittal or scaphocephaly, metoptic or trigonocephaly, coronal or anterior plagicephaly and lambdoid or posterior plagiocephaly).

The treatment is surgical from three to six months old and it is conditioned by the age when the diagnosis is done (which needs to be completed using radiographies, TAC and magnetic resonances), so these are the reasons for the importance of its early detection.

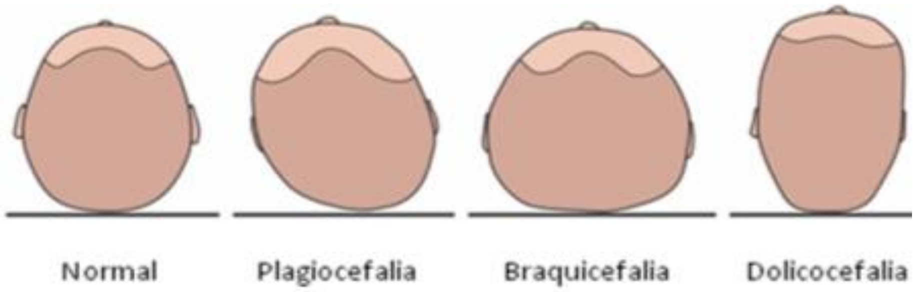
In the case of the branchycephaly and the postural plagiocephaly the cranial deformity is due to external pressures.

These ones are far more frequent, they affect up to 12% of babies and rising. It represents nearly 80% of all the cranial deformations.

The main cause is the recommendation done by paediatricians and nurses since 1992 when the American Academy of Paediatricians dictated the rule to make newborns sleep supine (on their backs) in order to avoid the “Sudden death of the nursling”.

If a child is quiet and his/her caregivers either hardly ever hold him on arms when he/she is awake or they do not change his/her posture, a flattening of the occipital zone is produced due to the constant pressure on the occipital zone producing a branchycephaly.

When a child, either lying supine or prone, always tilts his/her head on the same side, a lateral asymmetric flattering or plagiocephaly can appear, also due to the constant pressures on the same spot of the crane. In this case a torticollis (contracture of any of the neck muscles) has to be discarded as it could prevent the head from moving.



The consequences of a branchycephaly or a plagiocephaly which are not correctly treated may be aesthetic problems, orthodontic alterations, ocular alterations (exotropia) and delays on growth.

The treatment consists on a series of corrective postural measures, in addition, in case of torticollis the treatment consists on some exercises that should be done daily during the first months. When the babies turn six months old these treatments lose their effectiveness.

There are other treatments based on special pillows for its shape that minimize the pressure on specific places and corrective helmets that the baby has to wear during the first months but they are rarely used because of their high cost and their visual impact.

From here comes the importance of the correct and early diagnosis done by the health professionals (paediatricians and nurses) as well as the tracing of the evolution since they are able to train and support parents in their task.

Nowadays, the primary attention medical centres of our country own tape measure to do tracings of the cranial perimeter and when cranial deformities are noticed visually they are evaluated and the professional decides if corrective measures are applied or they are sent to the neurosurgeon. However, there was no reliable instrument to diagnose cranial dysmorphias.

The "Servei de Neurocirurgia" from the hospital "Sant Joan de Déu" led by Dr. J.Bosch Hugas and Dr. J. M. Costa Clara has recently designed a measuring tool with a classification index as well as a protocol to do a correct derivation.

The evaluation consists on putting a flexible ribbon placed around the crane embracing the maximum perimeter. On this ribbon key spots are marked taking the line of the nose as a reference. Then using a "craneometer" (a tool similar to a caliper) the distance between the key marks is measured. Nevertheless, just a few medical centres own a "craneometer" and the reliability of the tool still depends on the ability of the person who does the measuring.

2.1.1.The diagnosis

In order to diagnose precisely a cranial deformity two parameters are taken into account: the cefalometric index and the plagicephaly index.

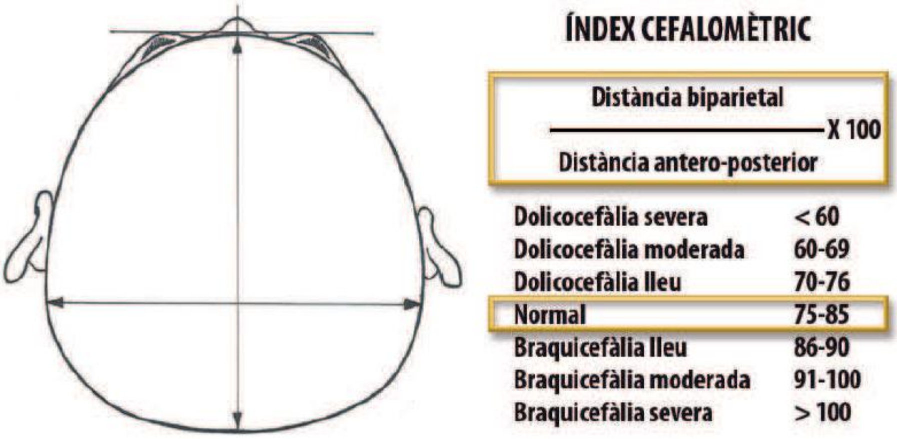
Cefalometric index (CI)

It is a numerical value that defines the elongation (scaphocephaly) or the flattening (branchycephaly) of the crane. It is the result of multiplying by 100 the result of the division of the maximum biparietal distance between the anteroposterior distance as seen in the picture. The 75-85 is the ideal value. Any inferior value shows the intensity of a scaphocephaly, while any higher value is a branchycephaly.

Having the cefalometric index deformations can be classified in:

Branchycephaly: mild(CI=86-90),moderate(CI=91-100) and severe(CI>100)

Scaphocephaly: mild(CI=70-76),moderate(CI=60-69) and severe(CI<60)



ÍNDEX CEFALOMÈTRIC

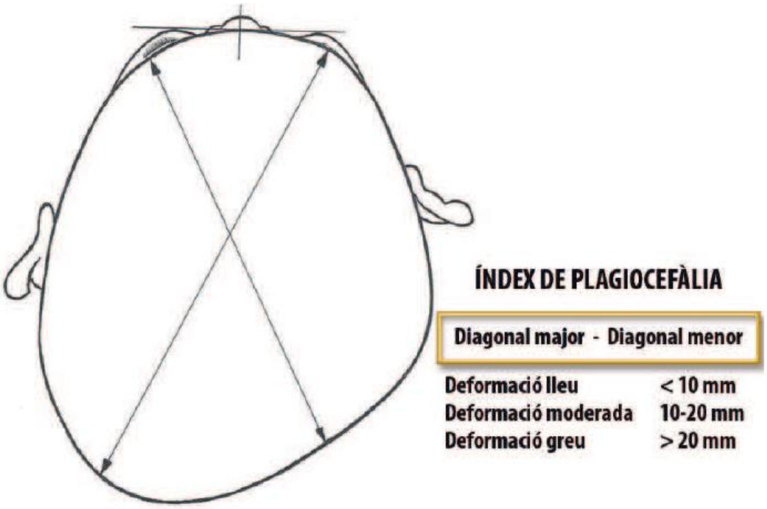
$$\frac{\text{Distància biparietal}}{\text{Distància antero-posterior}} \text{X 100}$$

Dolicocefàlia severa	< 60
Dolicocefàlia moderada	60-69
Dolicocefàlia lleu	70-76
Normal	75-85
Braquicefàlia lleu	86-90
Braquicefàlia moderada	91-100
Braquicefàlia severa	> 100

Plagicephaly index(PI)

This index evaluates the dis-symmetry between the major diagonal and the minor diagonal of the crane. It is considered mild if it is lower than 10mm, moderate if it is between 10 and 20 mm and severe if it is higher than 20mm.

As there is not a much extended way to do the measures, some scientists put a 30 degree gap between the vertical line and the diagonals.



ÍNDEX DE PLAGIOCEFÀLIA

Diagonal major - Diagonal menor

Deformació lleu	< 10 mm
Deformació moderada	10-20 mm
Deformació greu	> 20 mm

Source: Joan Bosch i Hugas, Josep Maria Costa i Clara*: La plagiocefàlia posicional: una tasca d'Atenció Primària*, Generalitat de Catalunya and Sant Joan de Déu

2.2.Interview to an expert[1]

Willing to get a better view of the environment of the primary attention diagnosis I interviewed Pilar Gussinyé Canaval.

She has a degree in medicine and surgery, specialised in paediatrics and familiar and community medicine. She works as a paediatrician in the “ABS Vall del Tenes, Bigues” and in the emergencies department in the “Hospital General de Granollers”.

How does the diagnose of cranial deformations work?

It is part of “The healthy kid” protocol.

In the regular revisions of a breastfeeding baby its cranial perimeter is measured and the crane shape is observed, if a deformation is visually detected, a monitoring is done to evaluate its evolution.

What is the treatment for a cranial deformation?

Once a deformation without improvement in the following weeks is detected, there has to be a distinction between a early closure of cranial sutures and a deformity caused by external pressures (which is more usual). This evaluation is done in a subjective way by observation. Given that there is a suspicion of early closure of sutures a radiography and/or a sonography are made, if there is a confirmation the case is derived to a neurologist and/or a neurosurgeon. If the deformation is a postural plagicephaly, recommendations are given to the parents in the consulting room or by the “CDIAP” (“Centre de Desenvolupament Infantil i Atenció Precoç”).

How often do you check the crane of each baby?

It is done in a formal way each month during the first six months of life and, afterwards, every three months until the kid is two years old.

In case of plagicephaly, a revision can be done every week until the objectives in the parents’ education are accomplished and/or the deformation improves.

[1] The original versión in Catalan of this interview can be found in the second annex.

How do you track the growth of a cranial deformation?

In the primary attention, it has been done visually until nowadays, that the “cranometer” is being distributed in some centres. However, there is no register or similar.

Do you have a tool to do an objective diagnose?

We have the “craneometer”. However, its use is not normalized yet and professionals are not formed institutionally.

What consequences does the inaccuracy of diagnoses have?

Low resolution in consulting room and an excessive derivation to “CDIAP”s

Do you think that an accurate method to diagnose cranial deformations would be useful?

Yes. It will allow primary attention professionals to do an objective evaluation of the evolution and it could also be positive for the parents as they could see the beneficial effects of the recommendations’ application.

Moreover, it would improve the efficiency of consulting rooms by decreasing the number of derivations to specialized services like the CDIAP.

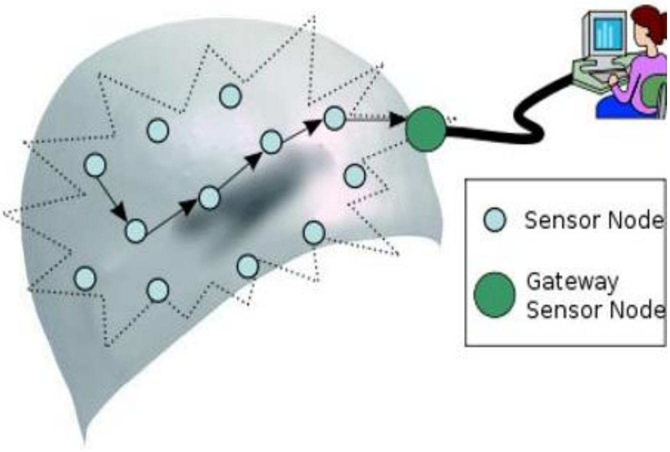
3.Practical scope

3.1.Evolution of the idea

During the process of doing the project the idea has changed in order to fit the real possibilities in my hands and so as to achieve my objectives.

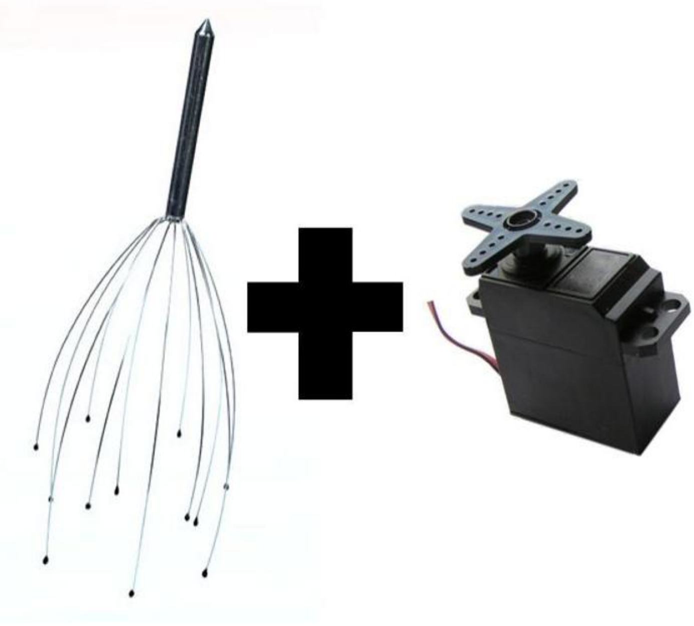
First idea

Firstly, I wanted to mix a swimming cap able to acquire the shape of the head with some sensors that would track its position and send it to a computer. Then the computer would show a detailed diagnosis of the cranial deformation. After some research I found what is called a wireless sensor network (WSN), but I rejected the idea as I considered it too difficult and out of my possibilities in terms of money and technology. As I lost the way to measure, I had to find another and here came the second idea.



Second idea

The other way to do the measurements is using servomotors which can track its position angle, so I thought about a structure with determined measures where I could easily know the distances required to diagnose cranial deformations. A good structure that fitted my conviction was a head massager.



The original concept was to create a direct USB interface between the servos and the computer. Nevertheless, I soon realized that the only way to link these two components was a microcontroller board so I had to add this feature to the design.

Third idea

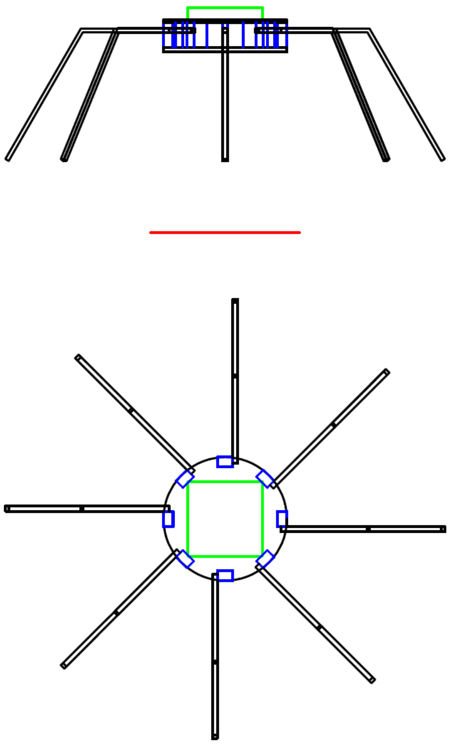
Having a good way to measure I focused on building the structure, when I realized that the shorter the mobile parts were the more accurate it would be. Therefore, I thought of a structure with more static parts than mobile parts. However, the structure that I designed was not handy enough and didn't convince me.

Otherwise, I realized of the need of a support which adds the reference of the crane so as to perform a better measuring.

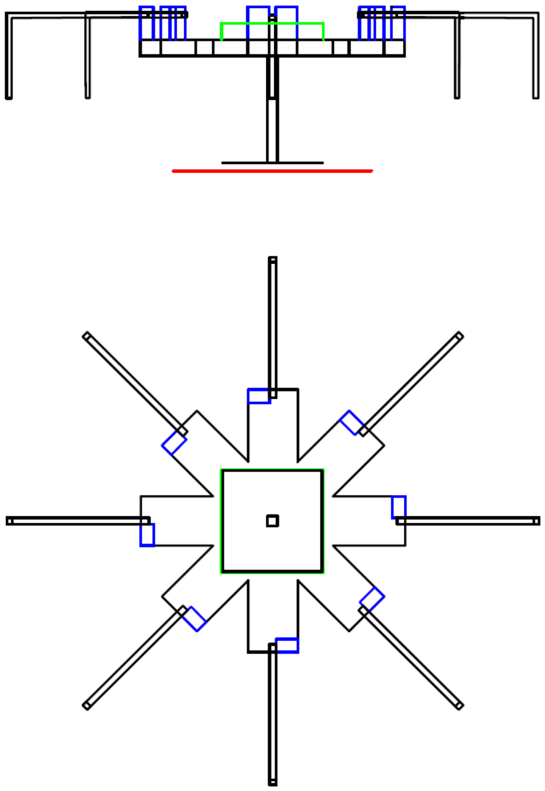
Final idea

Lastly, I came up with the definitive approximation by adding again the concept of something you wear on your head and designing the tool with a round shape.

Second idea structure map



Third idea structure map



3.2.Hardware

3.2.1.Introduction

At doing the practical experience of this research project I have worked with different material and I have followed a path to create the device.

3.2.2.List of components:

Electronic components:

-Arduino Duemilanove

-8 Sub-micro servos (Pololu reference: #1053)

-8 Force sensing resistors (force sensors) (Pololu reference: #1696)

- 4 Resistors (10KΩ)

-Several wires

Other components:

-Part of a plastic recipient

-8 Small plastic pieces

-LEGO pieces:

    -8 Cross bars

    -8 Connections number 6 (90 degrees angle)

    -8 Connections number 4 (135 degrees angle)

    -8 Round platforms

    -8 cross-round linkers

3.2.3.Construction

3.2.3.1.Mechanical part

Mobile arms

When building the mobile parts I realized there are some objectives to take into account:

-Making it adjustable to different sizes of head.

-Controlling the movement on the horizontal axis.

-Getting the maximum accuracy.

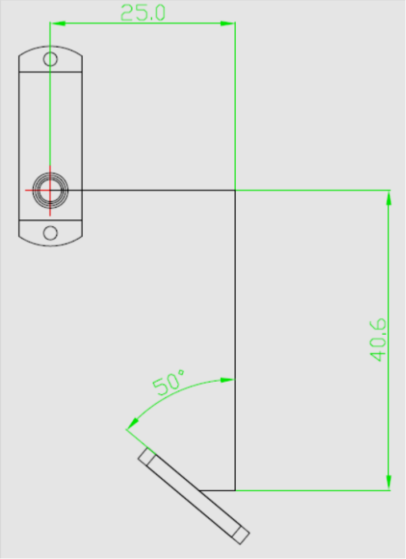
The main purpose was to suit any baby so I drew the approximate maximum and minimum perimeter to measure of a newborn's crane which are 450 millimetres and 300 millimetres respectively, to draw them I found their diameter: 143 mm (maximum) and 95 mm (minimum). Having this data I could find the horizontal distance to cover which is 23,9 millimetres.

Aiming to fit different sizes of head and measuring them in the same place I had the idea to do a curved arm and put the whole mobile part on the edge of the structure which would lead to do the measure higher on bigger heads and lower on smaller, therefore I decided to put a 90 degree angle curve near an edge of the arm.

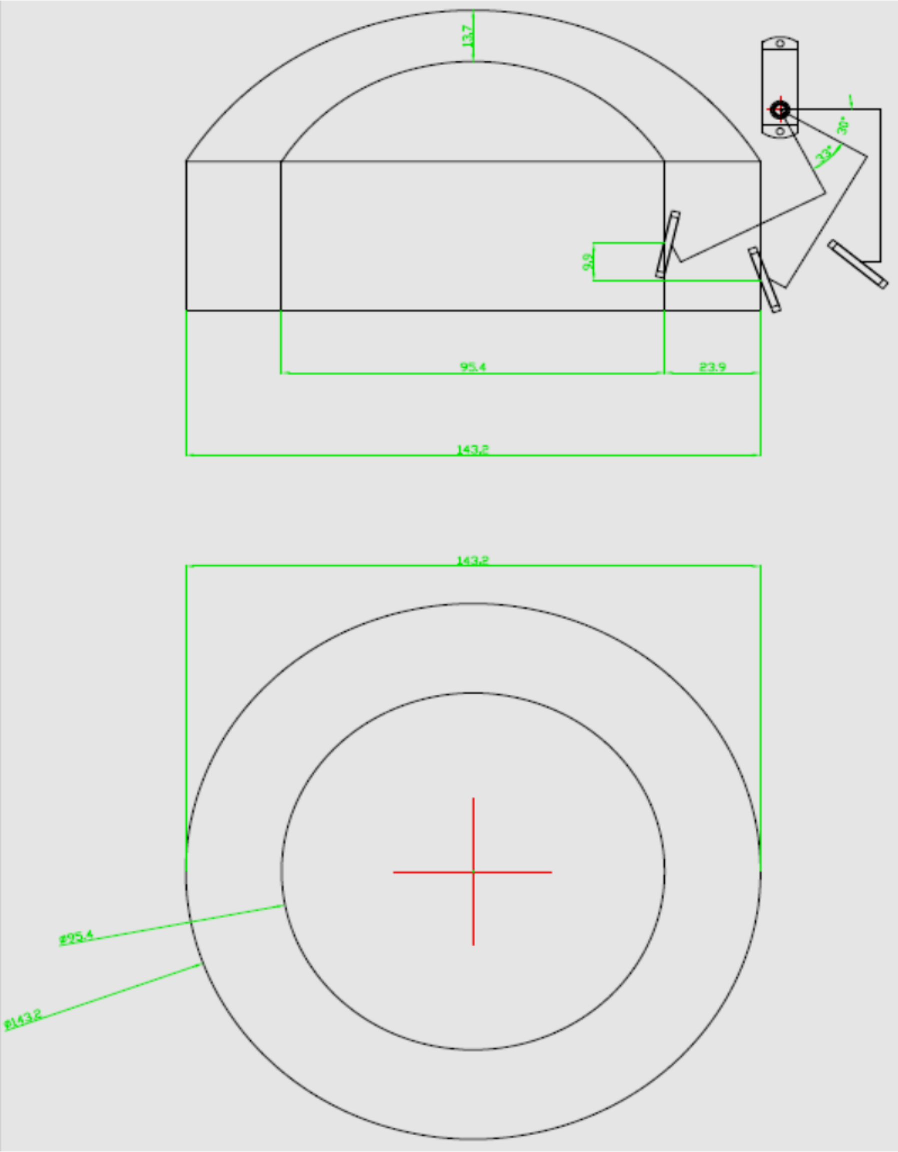
In conclusion, to gain accuracy the mobile arm should be as short as possible.

Having these references I proceeded to the design of the mobile arm:

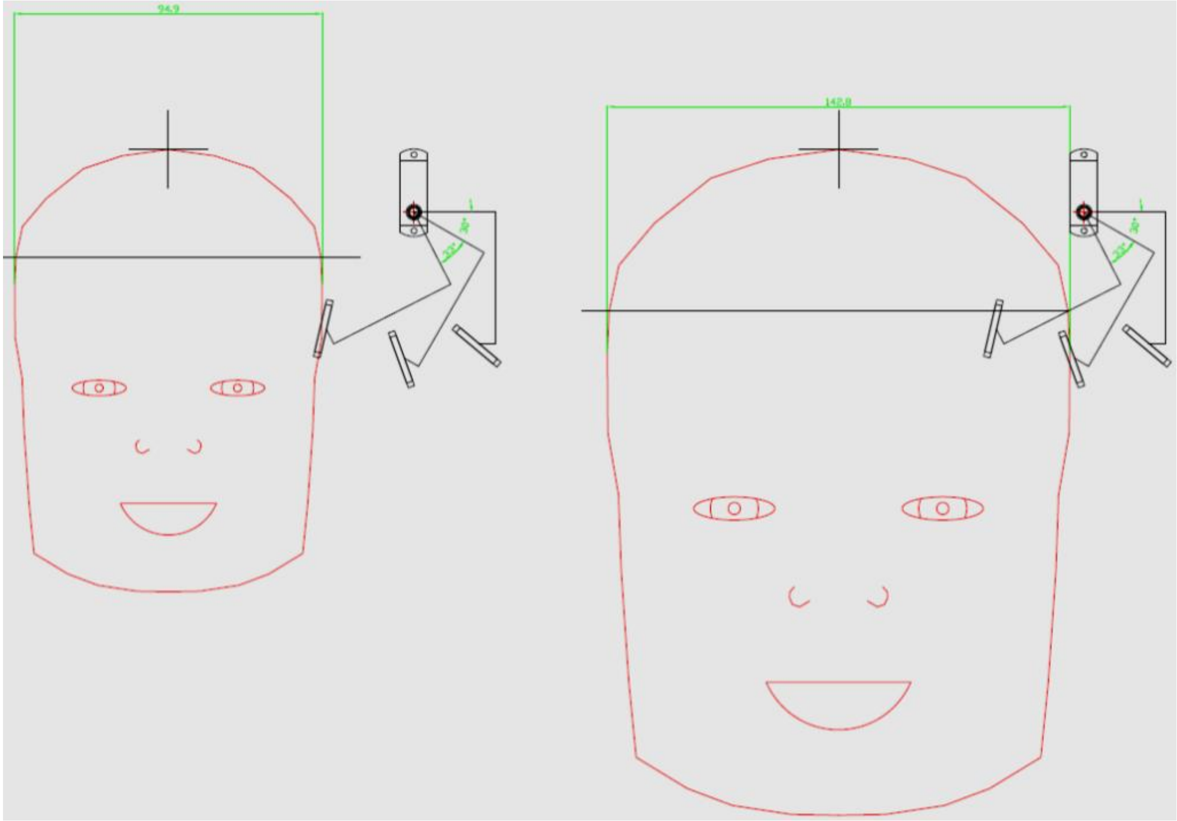
Mobile arm design



Final design adaptation to maximum and minimum sizes of a head: (plan and cross-section)



Final design adaptation to different cranes with the same position:

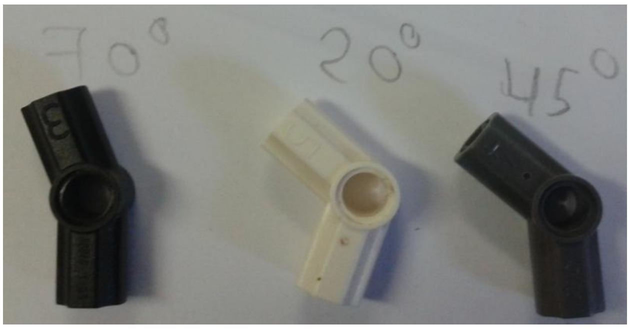


Having the design completed the next step was the construction:

In the first place, I wanted to give a shape to a metallic wire to do the arm structure but I soon realised it would be very difficult to stick it to the servomotor so I chose a plastic structure as it would be lighter and easier to stick.

The best way I found to build the arm was to use LEGO, though it has a rather informal appearance, it offers a comfortable and handy resource to do several identical structures which will not have different measures, in addition, it is easy to stick it anywhere using glue.

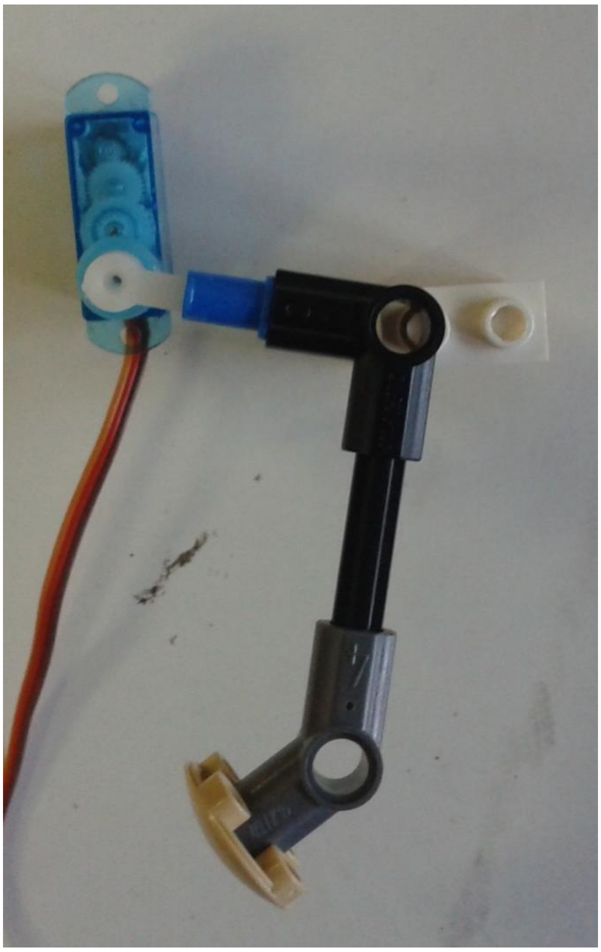
I encountered some difficulties as I aimed to a 50 degree angle in the structure and these were the pieces LEGO offers:

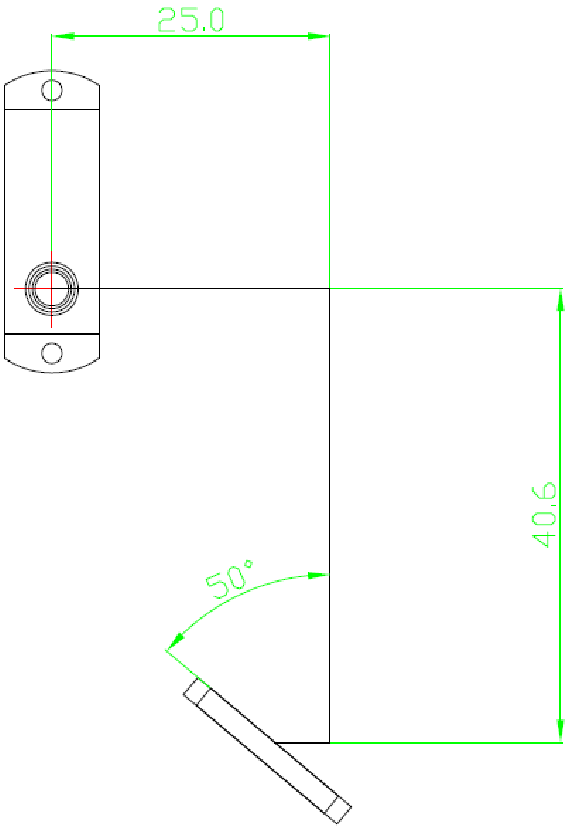


So I used the 45 degree angle connection doing a change to the design, however this change was not really relevant.

After finding all the pieces I attached them together to create this result:

Final result and design:





Main structure

To start with I did a list of features that the structure needed to have:

-Round shape.

-Diameter similar to the maximum baby's crane diameter (14,3 centimetres).

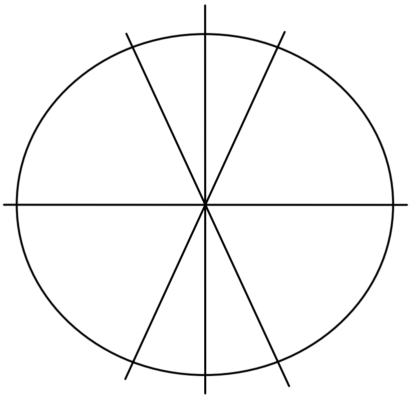
-It has to be supported on the newborn's head as a reference.

Searching at home I found a cylindrical hard plastic container with a 11,8 centimetres diameter.

I cut it to have a smaller structure to work with and this was the final result:

Besides, the cavity of this cylinder is a very good location to place the microcontroller with all its electronic components.

Attaching the mobile arms to the structure



Once I had both the structure and the mobile arms I had to put them together but first I had to know how. In order to measure the cefalometric index the vertical and the horizontal measures are needed and for the plagicephaly index the two diagonals which are 30 degrees far from the vertical axis are required. Consequently, I got a final result similar to the picture on the right. Despite the difficulty that the round shape of the structure supposed, I managed to get the right angle by using 8 small plastic pieces as a support for the servomotors.

Placing the microcontroller and the force sensors

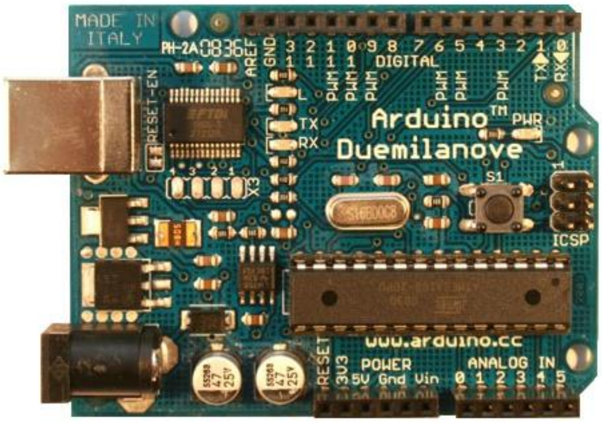
The final task was positioning the microcontroller and setting the force sensors; I placed the Arduino inside the cavity of the structure and I stack the sensors to the edge of the mobile arm and connecting them with the microcontroller with a wire.

3.2.3.2.Electronic part

The Arduino microcontroller

Arduino is an open-source single-board microcontroller platform extended around the world. It has the aim to ease the work in any type of electronic project. It consists of an open hardware with an Atmel AVR processor input output support. Besides, it has a language compiler and a boot loader.

In addition, it is programmed using a wiring-based language very close to “C” which allows anyone to get a close approach to standard programming.



The microcontroller board operates at 5V, has 14 digital input/output pins, 6 analogical inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It can be powered easily by a USB cable or by a AC-to-DC adapter or battery.

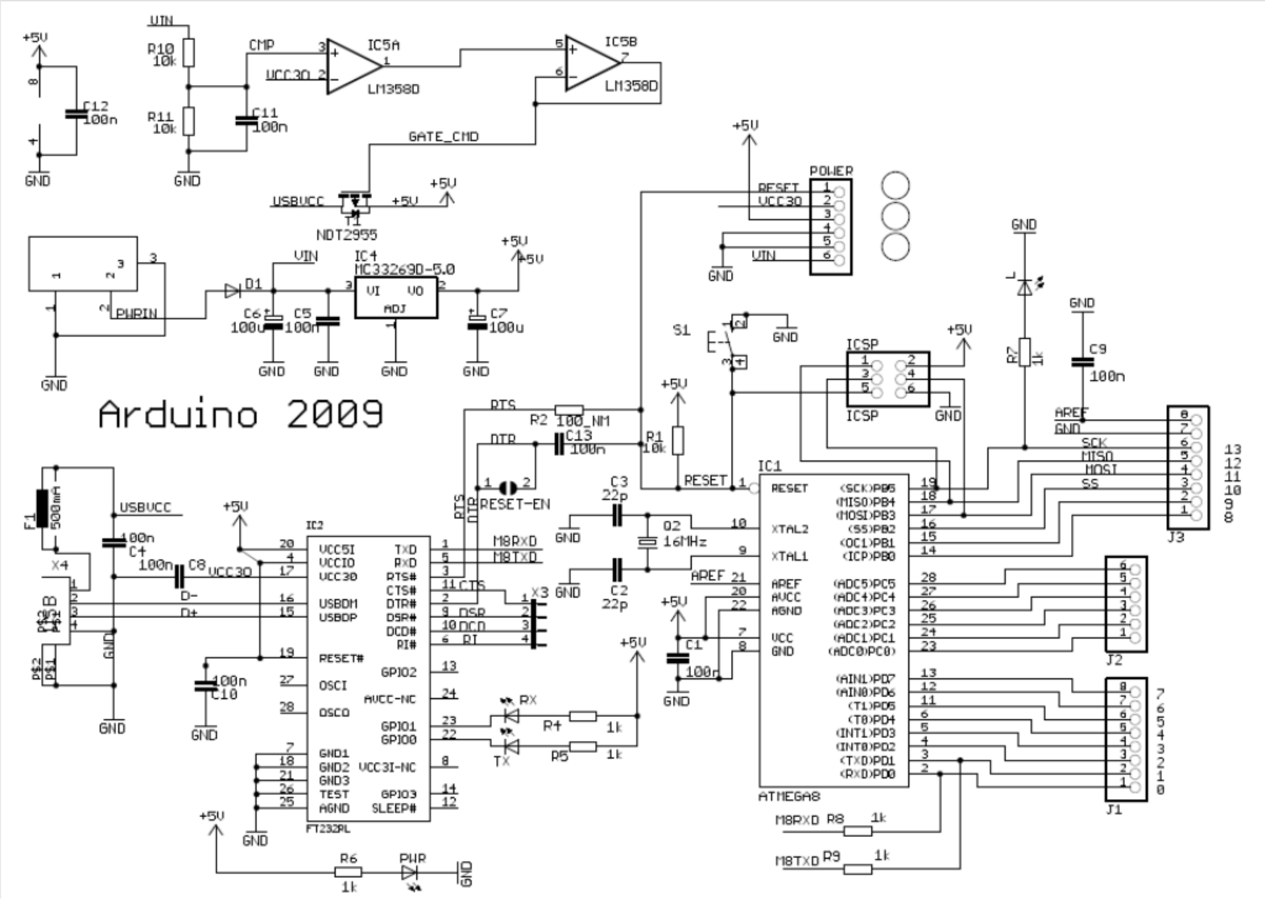
Inputs and outputs

Digital outputs: They are able to send intermittent voltage creating a signal that several devices can understand. I used these outputs to control the servomotors.

Analogical inputs: They measure the voltage wherever they are placed and map the signal between 0 and 1023. I used the analogical inputs to receive the information from the force sensors.

Digital inputs: They measure the voltage but they only give it the values of 0 or 1. They were ideal to plug a control button.

Arduino Duemilanove scheme



The sub-micro servos

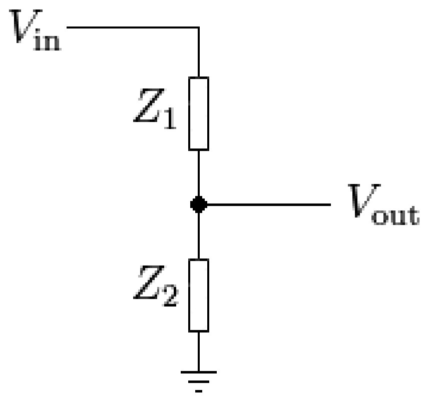
These sub-micro servos are one of the lightest servos, weighing just 3.7 g without the lead and they have the small dimension of 20.2 x 8.5 x 20.2 mm. They are intended for use around 4.8 V, I chose this sort of sensors because I didn't need strength and I wanted the servos to be light, small and handy.

All the servos need is receiving a signal from a digital output and being linked to the power supply. Therefore, they are connected to the negative terminal (or ground) by the brown wire, to the digital output by the orange wire and with the positive terminal by the red wire.

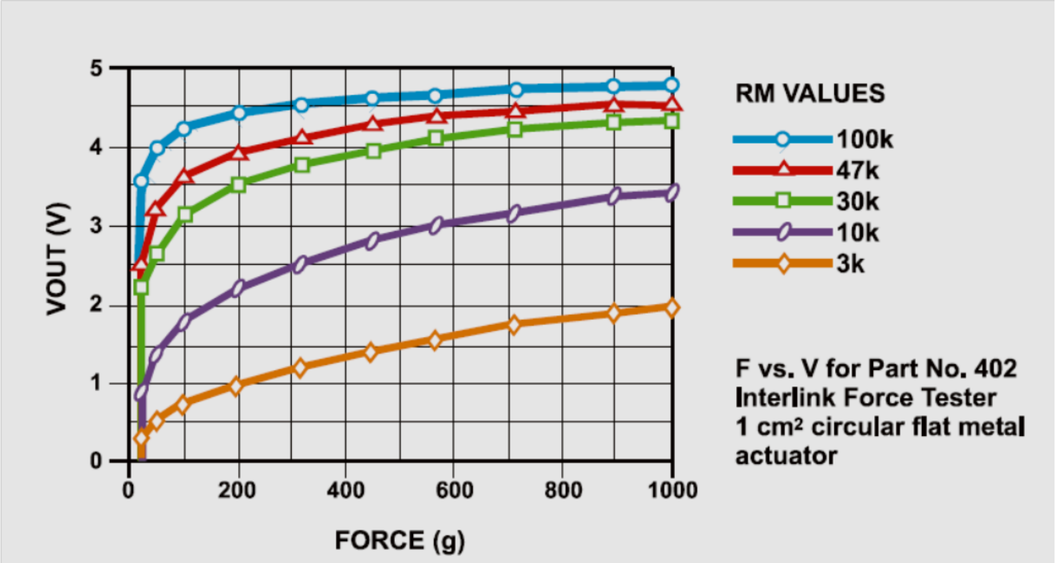
The force sensing resistors (sensors)

Force Sensing Resistors (FSR) are a polymer thick film (PTF) device which exhibits a decrease in resistance with an increase in the force applied to the active surface. Its force sensitivity is optimized for use in human touch control of electronic devices.

As the force sensors work as a variable resistor I needed to put a voltage divisor in order to get a signal from the sensors. A voltage divisor is a linear circuit that allows you to get an output voltage, it consists of two resistors in series. Regarding the picture, I put the positive terminal as V_{in}, the sensor as Z_1, the analogical input as V_{out}, a normal resistor as Z_2 and lastly the negative terminal as the ground (the three horizontal lines).

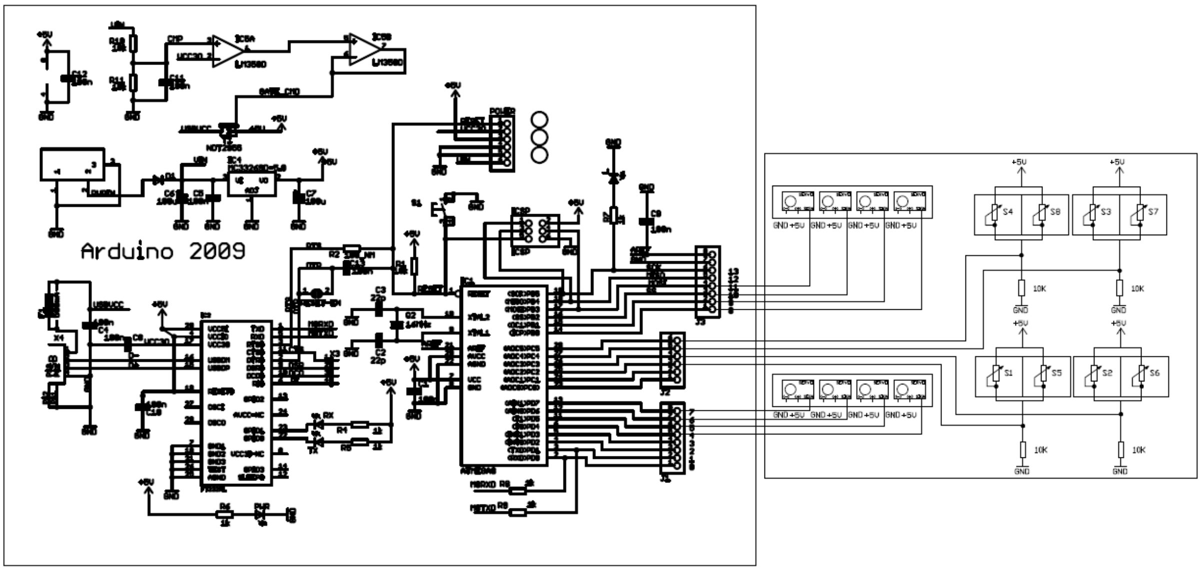


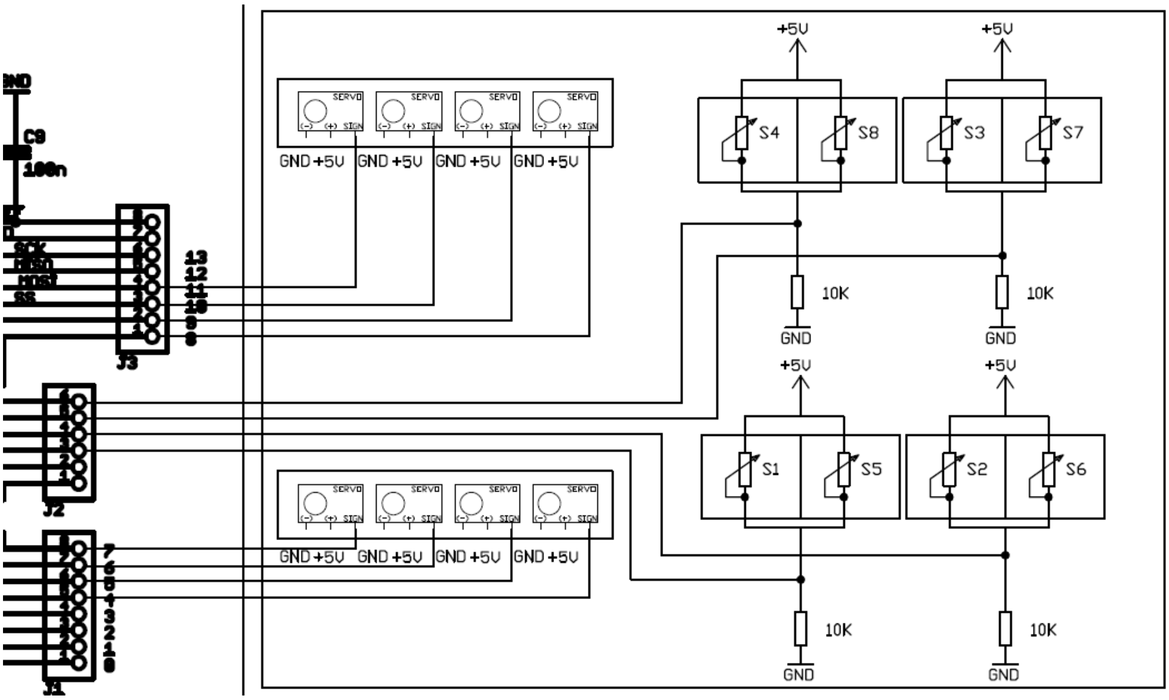
So as to know which resistor I had to put as Z_2 to get a correct reading I searched in the force sensing resistor datasheet, there I found a graphic that shows the behaviour of the sensor put in a voltage divisor with different resistors (RM). I chose to put a 10 kΩ resistor as it has the most balanced behaviour in the graphic.



The microcontroller has only 6 analogical inputs, however I had to connect 8 sensors to them so I had to find a way to do so. After thinking some time I found a really simple solution for my problem; putting two variable resistors in parallel. This solution achieved my objective but it changes the signal that receives the input, however this can easily be fixed by changing the tolerance value in the program. Putting the two variable resistors in parallel brought me a restriction as well; the measurement with each sensor must not be simultaneous as there would be interferences between their measurements.

Final connections scheme:





3.2.4.Model of a baby's crane

To test the prototype of my measuring tool I got a baby doll, however its crane was too small. As I wanted a closer approach to a real crane of newborn size I created a clay model with the correct size.

3.3.Software

3.3.1.Programming

Programming is using an artificial language designed to communicate instructions to a machine, particularly a computer. Programming languages can be used to create programs that control the behaviour of a machine and/or to express algorithms precisely.

To achieve this objective the programmer has to learn a programming language and use it to communicate some instructions to an “electronic brain” which is usually a computer or any other electronic device able to understand this type of messages.

3.3.2.Programming with Arduino

So as to program my work I used the Arduino language as it is similar to some languages I already knew and it is a direct link to the microcontroller (Arduino Duemilanove).

The Arduino language is very similar to C or C++ and generally to the most common programming languages. In addition, it is used with the microcontroller by default so there is no need to find how to use it with Arduino Duemilanove.

Arduino programs can be divided into three main parts:

Structure: Signs or words that organise the program. Examples :(+ , ; , for)

Values: Numbers or words used in most of the actions. Examples :(int , true , boolean)

Functions: Actions done by the program. Examples :(analogWrite() , delay() , .write())

So as to understand it better we can assimilate structure to syntax, values to words and functions to verbs; regarding the features of normal languages.

3.3.3.Learning

It took me some time to learn the ropes in Arduino programming so I did some programs in order to have a step-by-step guide of my learning process.

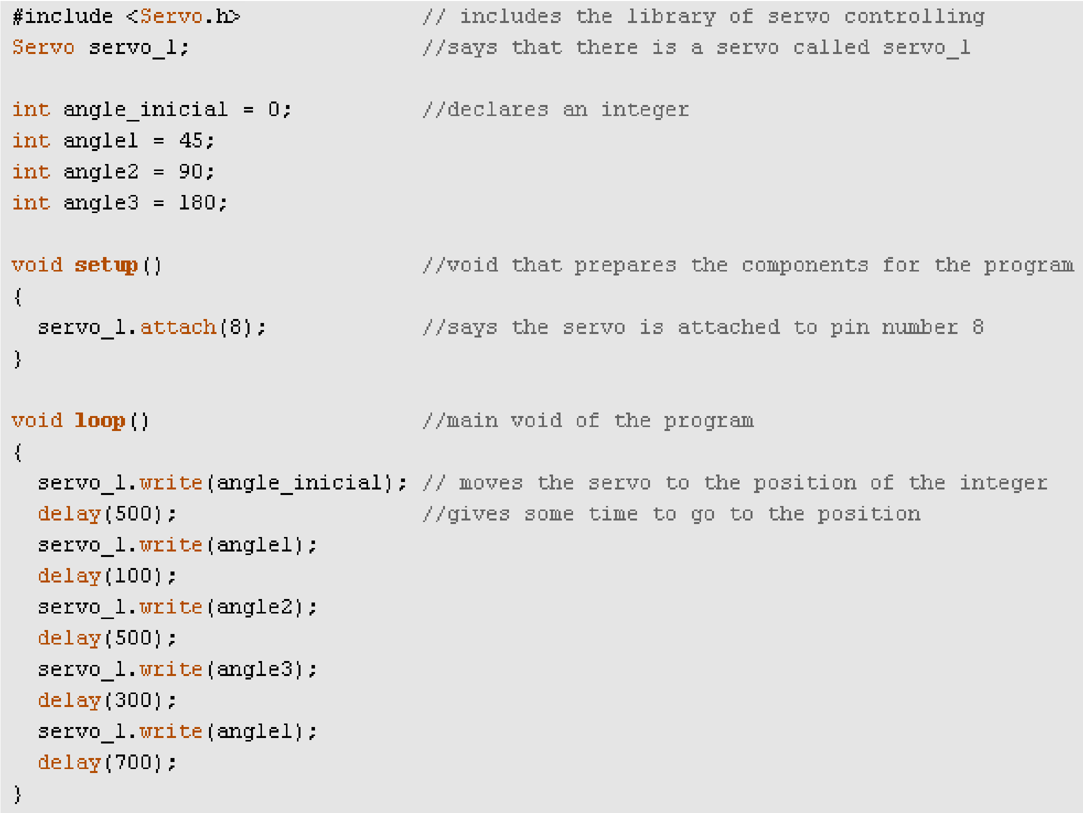
Test 1: Moving a servo

Objectives:

-Use the main structure (void setup, void loop)

-Add a library

-Test a servo (accuracy and general performance)



```
#include <Servo.h>                 // includes the library of servo controlling
Servo servo_1;                     //says that there is a servo called servo_1

int angle_inicial = 0;             //declares an integer
int angle1 = 45;
int angle2 = 90;
int angle3 = 180;

void setup()                       //void that prepares the components for the program
{
  servo_1.attach(8);               //says the servo is attached to pin number 8
}

void loop()                        //main void of the program
{
  servo_1.write(angle_inicial); // moves the servo to the position of the integer
  delay(500);                      //gives some time to go to the position
  servo_1.write(angle1);
  delay(100);
  servo_1.write(angle2);
  delay(500);
  servo_1.write(angle3);
  delay(300);
  servo_1.write(angle1);
  delay(700);
}
```

Result: The servo moves properly at different speeds and shows a pretty good accuracy at doing the angles.

Test 2: Control structures

One of the most important parts of programming is the control structures which allow microcontrollers to go further and “think”. Here I wanted to test a useful control structure: the “for”. It works as loop carefully controlled where you can choose the starting point, the condition and the change.

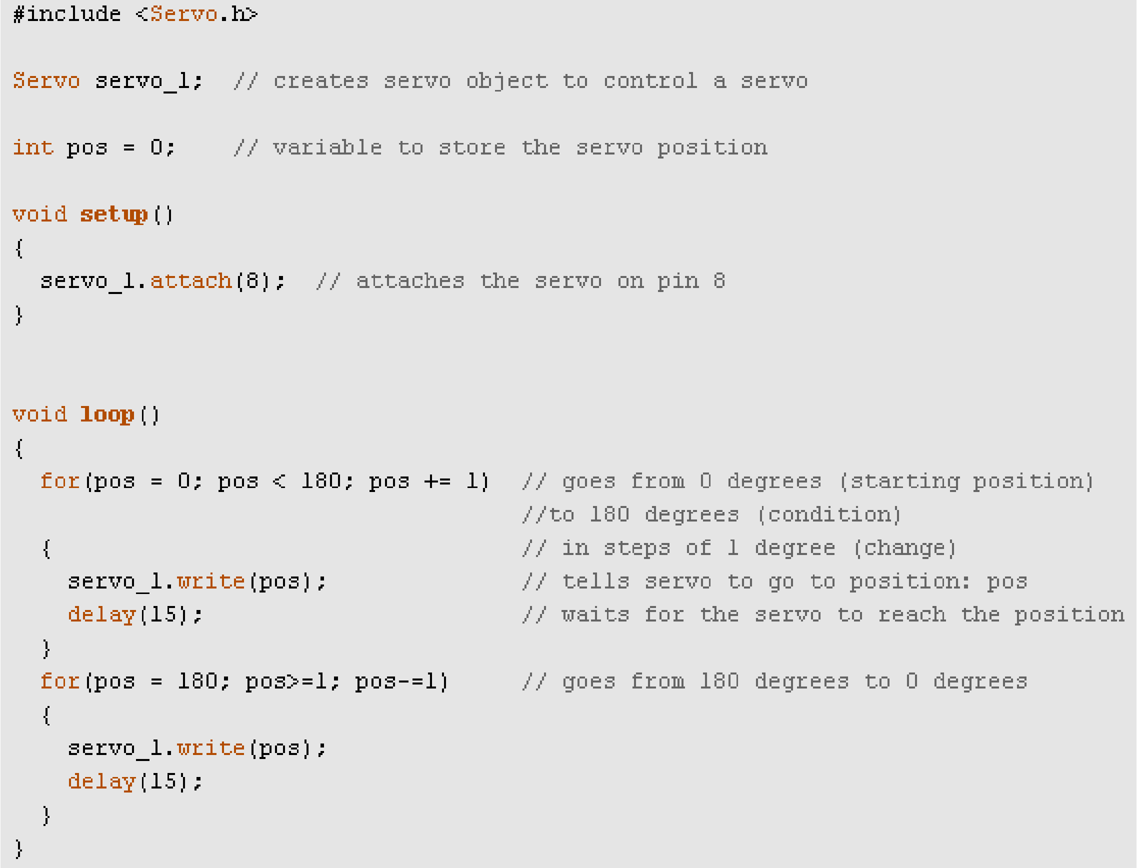
For structure:

for(“starting point”,”condition to check”,”change to do every time”)

{

“things to do in the loop”

}

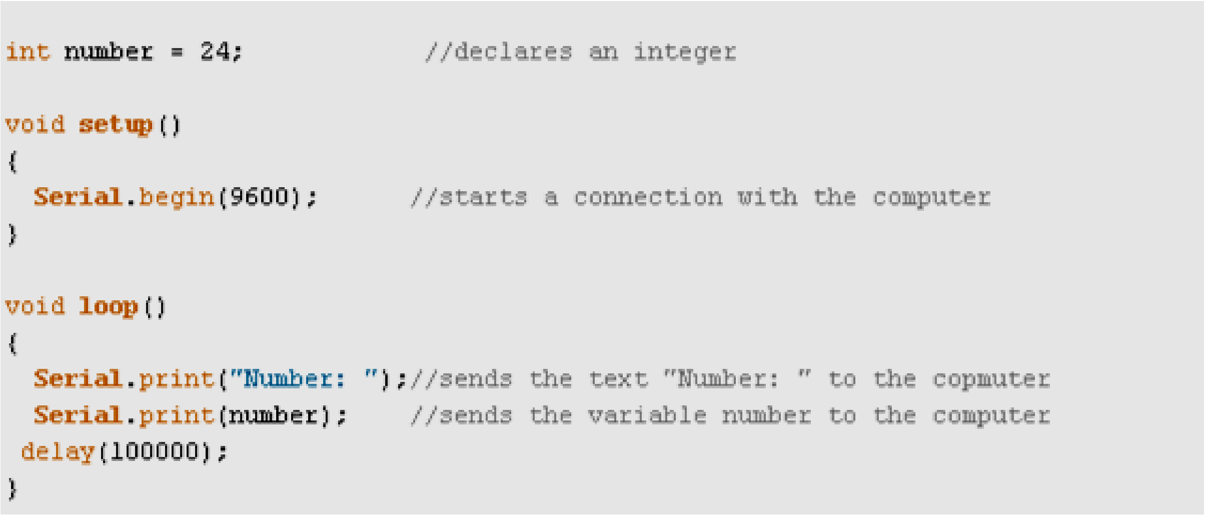


```
#include <Servo.h>

Servo servo_1;  // creates servo object to control a servo

int pos = 0;    // variable to store the servo position

void setup()
{
  servo_1.attach(8);  // attaches the servo on pin 8
}


void loop()
{
  for(pos = 0; pos < 180; pos += 1)  // goes from 0 degrees (starting position)
                                     //to 180 degrees (condition)
  {                                  // in steps of 1 degree (change)
    servo_1.write(pos);              // tells servo to go to position: pos
    delay(15);                       // waits for the servo to reach the position
  }
  for(pos = 180; pos>=1; pos-=1)     // goes from 180 degrees to 0 degrees
  {
    servo_1.write(pos);
    delay(15);
  }
}
```

Result: The servo goes gradually to 180 degrees and comes back to 0 degrees.

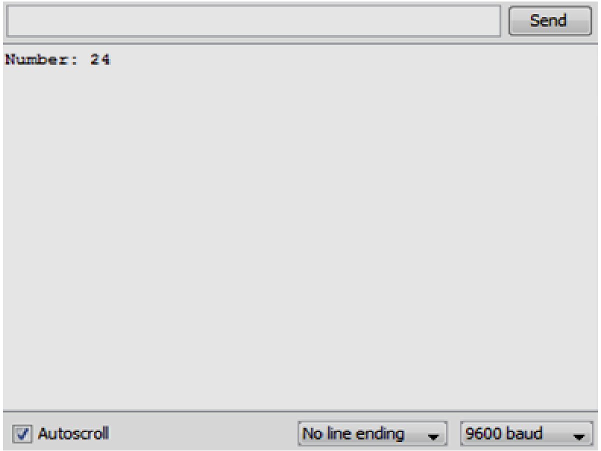
Test 3: Communication with the computer

Arduino is able to communicate with a computer via USB easily, to do it I use the function Serial which sends or receives information to/from a port previously chosen. I will use it to send a variable and a text to the computer.



```
int number = 24;            //declares an integer

void setup()
{
  Serial.begin(9600);      //starts a connection with the computer
}

void loop()
{
  Serial.print("Number: ");//sends the text "Number: " to the copmuter
  Serial.print(number);    //sends the variable number to the computer
 delay(100000);
}
```

Result: the message is correctly shown on screen:

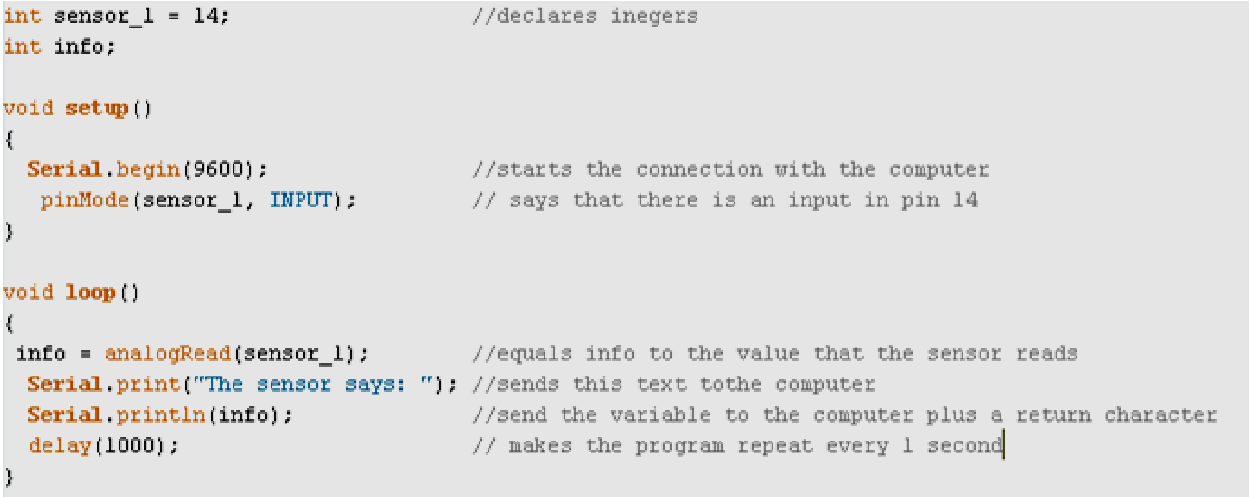


```
Number: 24
```

☑ Autoscroll   No line ending   9600 baud

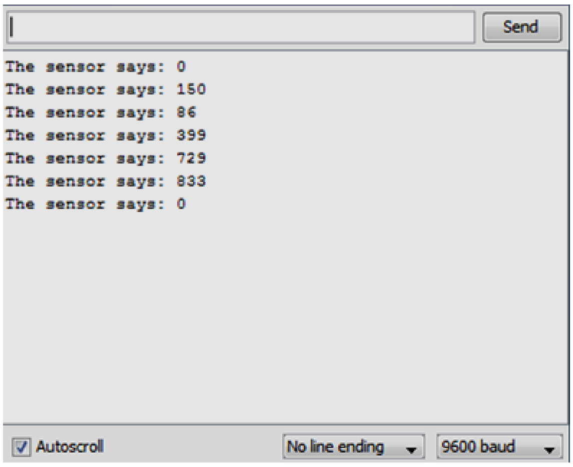
Test 4: Reading a sensor

Reading information from an input is vital for any project that needs interaction with the environment. Now I will test the force sensor used in my project by sending to the computer the information it gathers.



```
int sensor_1 = 14;                  //declares inegers
int info;

void setup()
{
  Serial.begin(9600);               //starts the connection with the computer
   pinMode(sensor_1, INPUT);        // says that there is an input in pin 14
}

void loop()
{
 info = analogRead(sensor_1);       //equals info to the value that the sensor reads
  Serial.print("The sensor says: "); //sends this text tothe computer
  Serial.println(info);             //send the variable to the computer plus a return character
  delay(1000);                      // makes the program repeat every 1 second
}
```

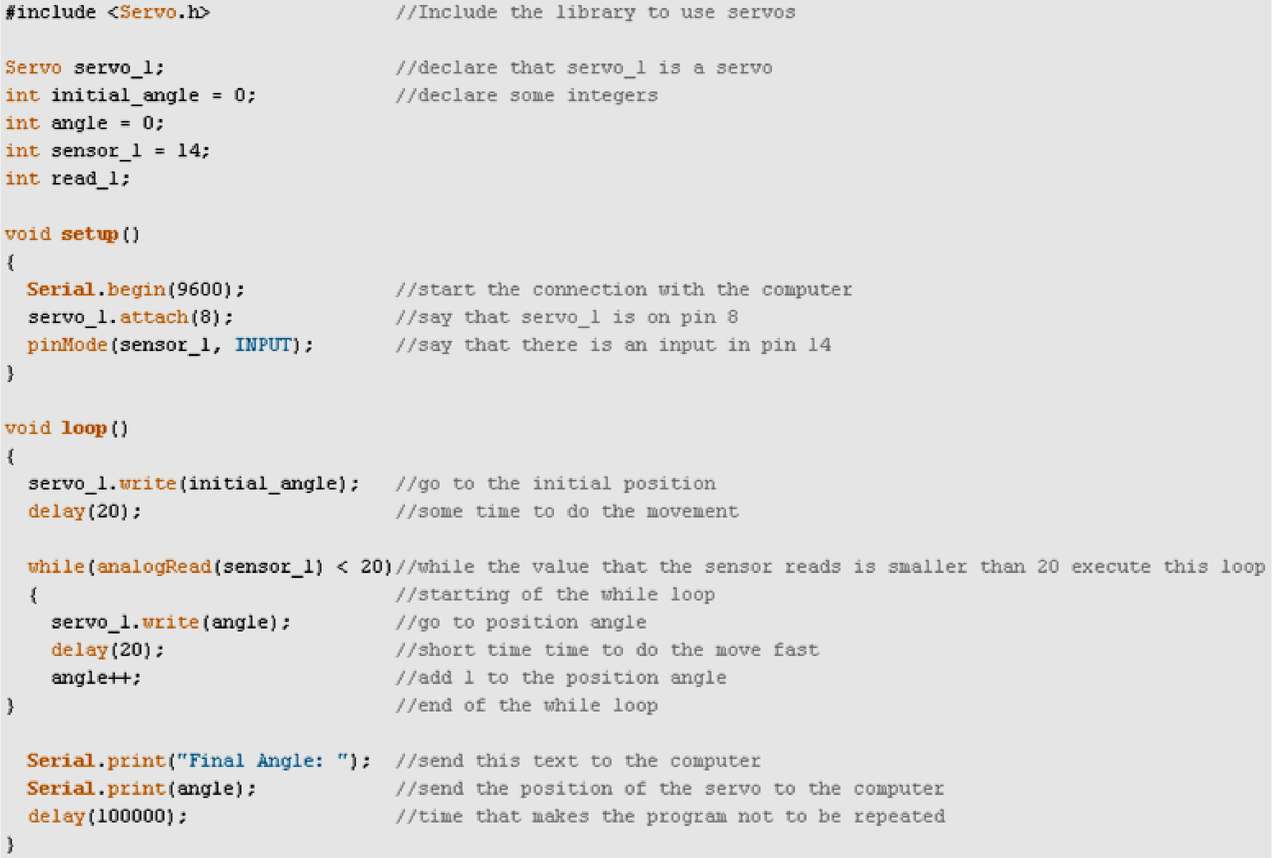
Result: The computer shows a value that changes depending on the force applied to the sensor:



```
The sensor says: 0
The sensor says: 150
The sensor says: 86
The sensor says: 399
The sensor says: 729
The sensor says: 833
The sensor says: 0
```

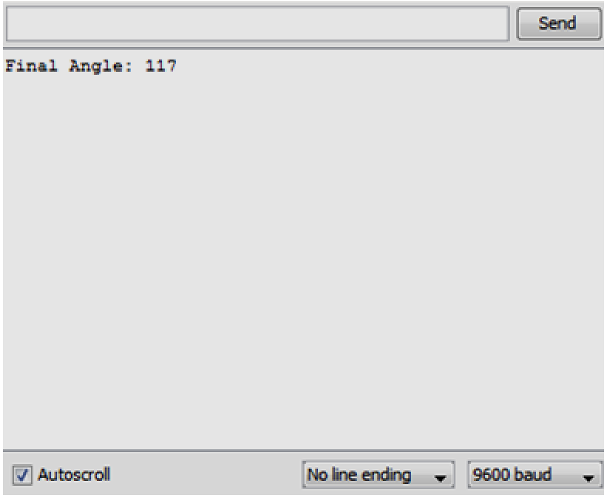
Test 5: Mix

Now that I have learnt what I need to develop a program to use in the project, I will do a synthesized version of the final program to test what I have learnt.



```
#include <Servo.h>                //Include the library to use servos

Servo servo_1;                    //declare that servo_1 is a servo
int initial_angle = 0;            //declare some integers
int angle = 0;
int sensor_1 = 14;
int read_1;

void setup()
{
  Serial.begin(9600);             //start the connection with the computer
  servo_1.attach(8);              //say that servo_1 is on pin 8
  pinMode(sensor_1, INPUT);       //say that there is an input in pin 14
}

void loop()
{
  servo_1.write(initial_angle);   //go to the initial position
  delay(20);                      //some time to do the movement

  while(analogRead(sensor_1) < 20)//while the value that the sensor reads is smaller than 20 execute this loop
  {                               //starting of the while loop
    servo_1.write(angle);         //go to position angle
    delay(20);                    //short time time to do the move fast
    angle++;                      //add 1 to the position angle
}                                 //end of the while loop

  Serial.print("Final Angle: ");  //send this text to the computer
  Serial.print(angle);            //send the position of the servo to the computer
  delay(100000);                  //time that makes the program not to be repeated
}
```

Result: In the first place, the servo moves to the starting position, then is starts moving until the sensor is pressed, as the sensor is pressed the final angle appears on screen:



```
Final Angle: 117
```

3.3.4.Final code

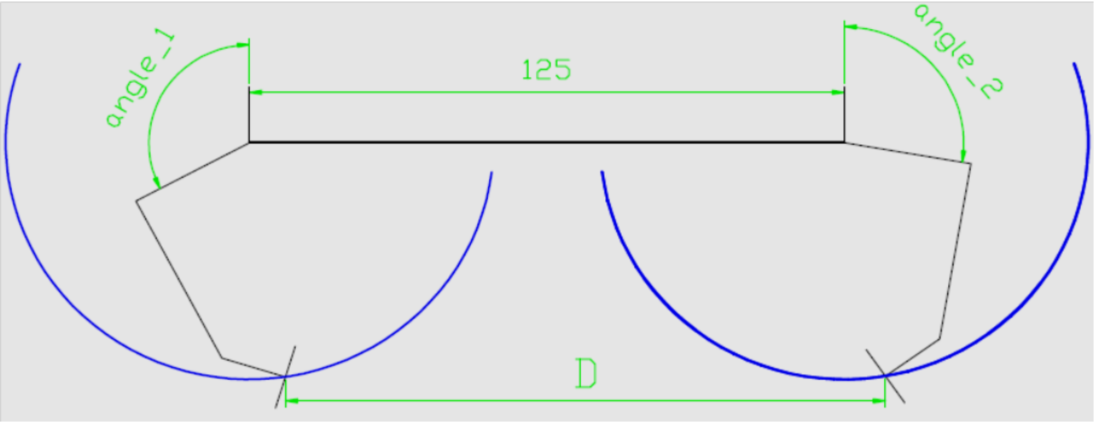
After learning and practising, I was capable of facing the challenging task of the final program. Nevertheless, I had to take smaller step before going for the definitive program.

3.3.4.1.Attempt of final program

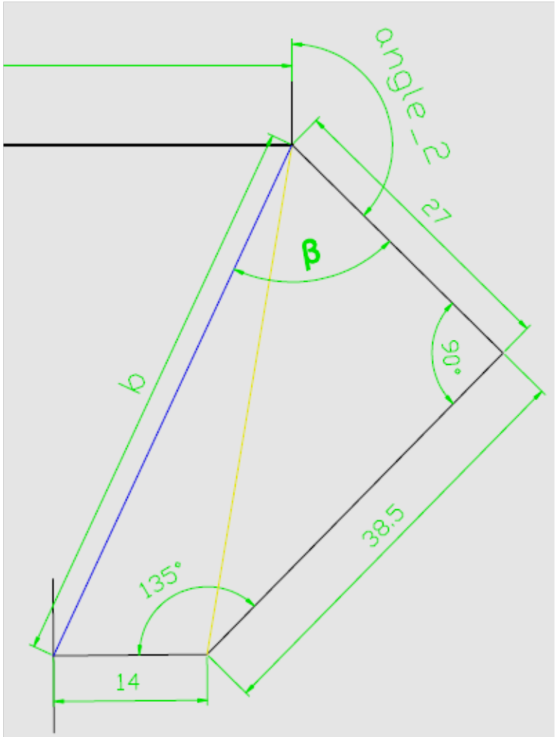
In this final test I want to measure the distance between the edges of two arms, therefore, I have to do the previous trigonometric calculus so as to get the distance between the arms directly from the servomotors' angle.

3.3.4.1.1.Calculus:

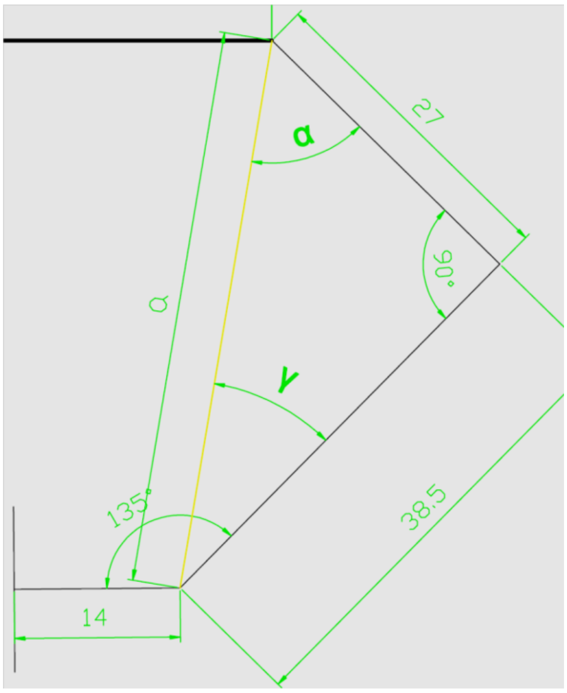
Firstly, I made a drawing to understand the problem; it shows that what I want to get is the distance between the arms (D) processing the position of the servo (angle_1 and angle_2).



After getting my ideas clear, I decided to simplify the structure by searching the distance between the servomotor and the edge of the arm (b) and the angle “β” shown in the drawing.



In order to know “b” and “β” I should first know the measures of the triangle inside the structure that appears in the next drawing. As it is a right-angled triangle I could know distance “a” by applying the Pythagorean theorem which says that the area of the square whose side is the hypotenuse (the side opposite the right angle) is equal to the sum of the areas of the squares whose sides are the two legs (the two sides that meet at a right angle) or “$a^2 + b^2 = c^2$” being ”c” the longest edge (hypotenuse).



Additionally, being a right-angled triangle also allowed me to use a rule of basic trigonometry, it states that the tangent is the ratio of the opposite leg to the adjacent leg or “ $\tan A = \frac{\text{opposite edge}}{\text{adjacent edge}}$ “

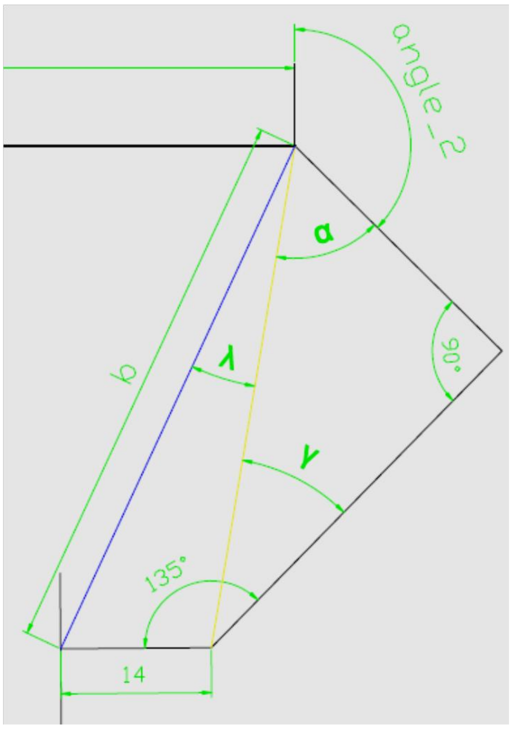
Calculation:

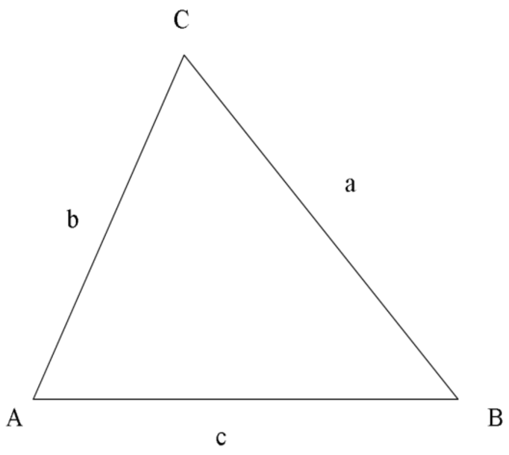
$$a = \sqrt{27^2 + 38.5^2} = 47.02mm$$

$$\alpha = \tan^{-1}\frac{38.5}{27} = 54.96°$$

$$\gamma = \tan^{-1}\frac{27}{38.5} = 35.04°$$

The next step to take was discovering “b” and “λ” so I focused in the second triangle inside the structure, as it is not right-angled I could not use the same methods as previously. However, I could overcome this obstacle as I know the Law of cosines that works in any triangle. This law is an extension of the Pythagorean theorem and establishes this equation: ” $c^2 = a^2 + b^2 - 2ab\cos C$“ and consequently: “ $C = \cos^{-1}\frac{a^2+b^2-c^2}{2ab}$ “ taking into account the drawing below.





Calculation:

$$b = \sqrt[2]{14^2 + 47.02^2 - 2 \cdot 14 \cdot 47.02 \cdot \cos(135 - 35.04)} = 51.33mm$$

$$\lambda = \cos^{-1}\frac{51.33^2 + 47.02^2 - 14^2}{2 \cdot 51.33 \cdot 47.02} = 15.58°$$

Afterwards, I only had to build the equation that gives me the distance between arms “D” from “angle_1” and “angle_2”, this step was far easier as I already had the angles “λ” and “α” and distance “d“. I started calculating “β” and then I set up the final formula where I used the trigonometry basics again.

Calculation:

$$\beta = \alpha + \lambda = 15.08 + 54.96 = 70.04°$$

$$D = 125 + 51.33\,(\sin(angle\_1 + 70.04) + \sin(angle\_2 + 70.04))$$

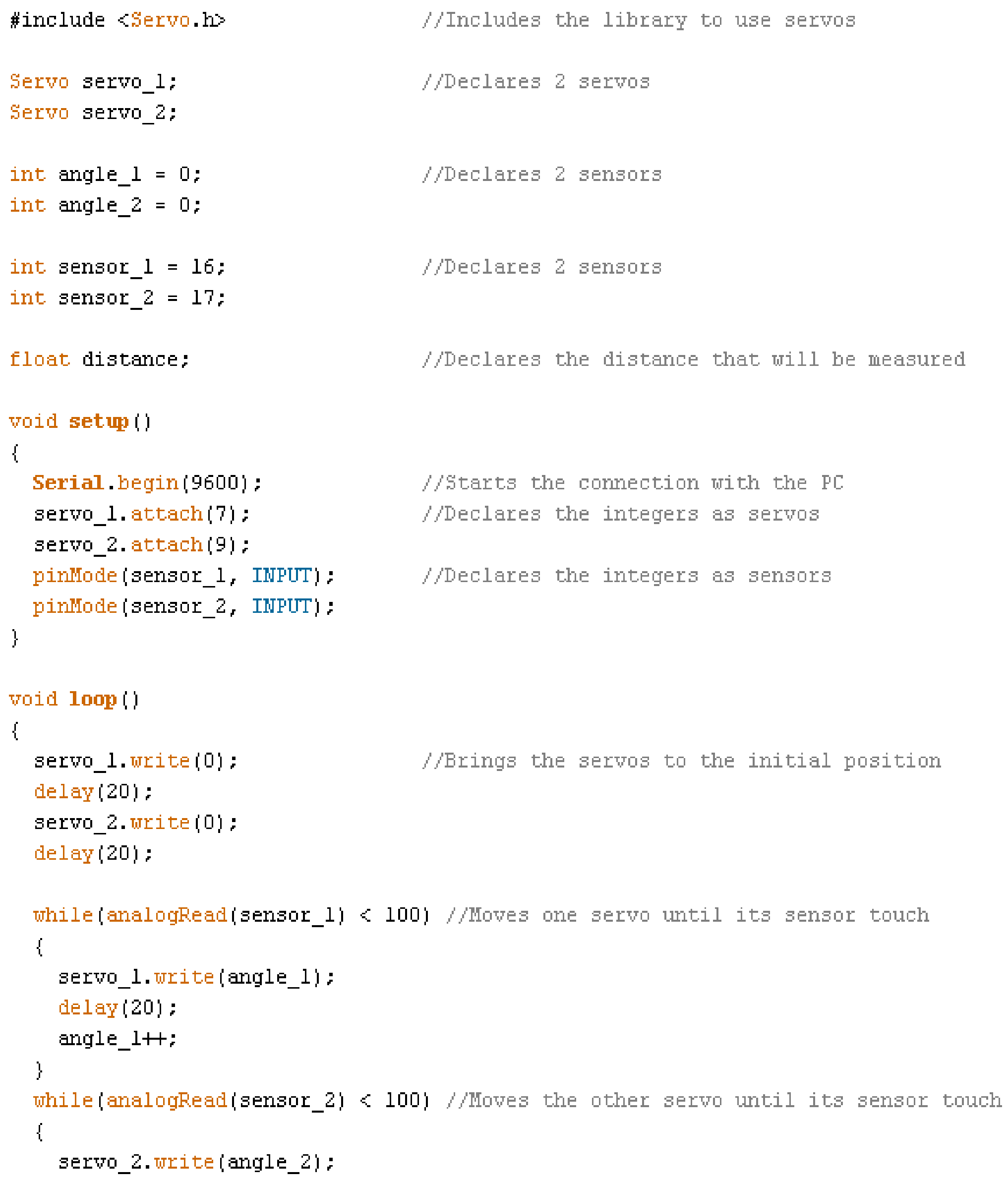
When I checked the Arduino functions I realized that the sinus works with radians so I had to change the unities of the angles. Conversion: $degrees \times \frac{\pi}{180} = radians$

Calculation:

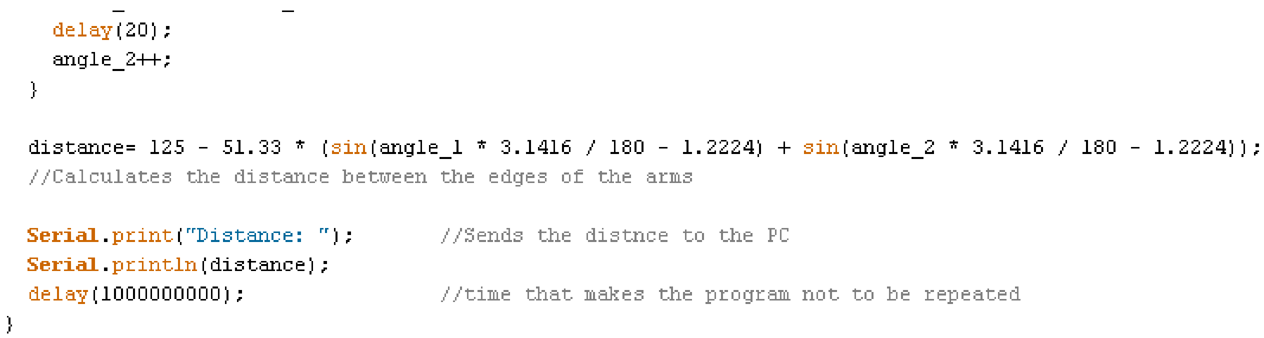
$$70.04° \cdot \frac{\pi}{180} = 1.2224\, rad$$

$$D = 125 + 51.33\,(\sin(angle\_1 \cdot \frac{\pi}{180} + 1.2224) + \sin(angle\_2 \cdot \frac{\pi}{180} + 1.2224))$$

With the equation I was able to build up the program:

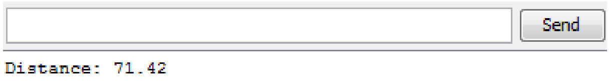


```
#include <Servo.h>                //Includes the library to use servos

Servo servo_1;                    //Declares 2 servos
Servo servo_2;

int angle_1 = 0;                  //Declares 2 sensors
int angle_2 = 0;

int sensor_1 = 16;                //Declares 2 sensors
int sensor_2 = 17;

float distance;                   //Declares the distance that will be measured

void setup()
{
  Serial.begin(9600);             //Starts the connection with the PC
  servo_1.attach(7);              //Declares the integers as servos
  servo_2.attach(9);
  pinMode(sensor_1, INPUT);       //Declares the integers as sensors
  pinMode(sensor_2, INPUT);
}

void loop()
{
  servo_1.write(0);               //Brings the servos to the initial position
  delay(20);
  servo_2.write(0);
  delay(20);

  while(analogRead(sensor_1) < 100) //Moves one servo until its sensor touch
  {
    servo_1.write(angle_1);
    delay(20);
    angle_1++;
  }
  while(analogRead(sensor_2) < 100) //Moves the other servo until its sensor touch
  {
    servo_2.write(angle_2);
```



```
    delay(20);
    angle_2++;
  }

  distance= 125 - 51.33 * (sin(angle_1 * 3.1416 / 180 - 1.2224) + sin(angle_2 * 3.1416 / 180 - 1.2224));
  //Calculates the distance between the edges of the arms

  Serial.print("Distance: ");       //Sends the distnce to the PC
  Serial.println(distance);
  delay(1000000000);                //time that makes the program not to be repeated
}
```

Result: The servos go to the starting position, then one starts moving forward and stops when the sensor touches something, immediately, the other servo starts moving and stops when its sensor is pressed, afterwards, the value of the distance in millimetres measured appears on screen.

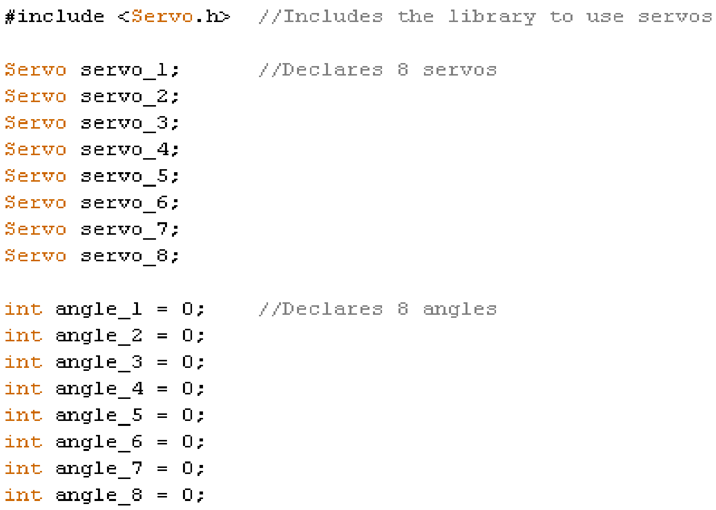


```
Distance: 71.42
```

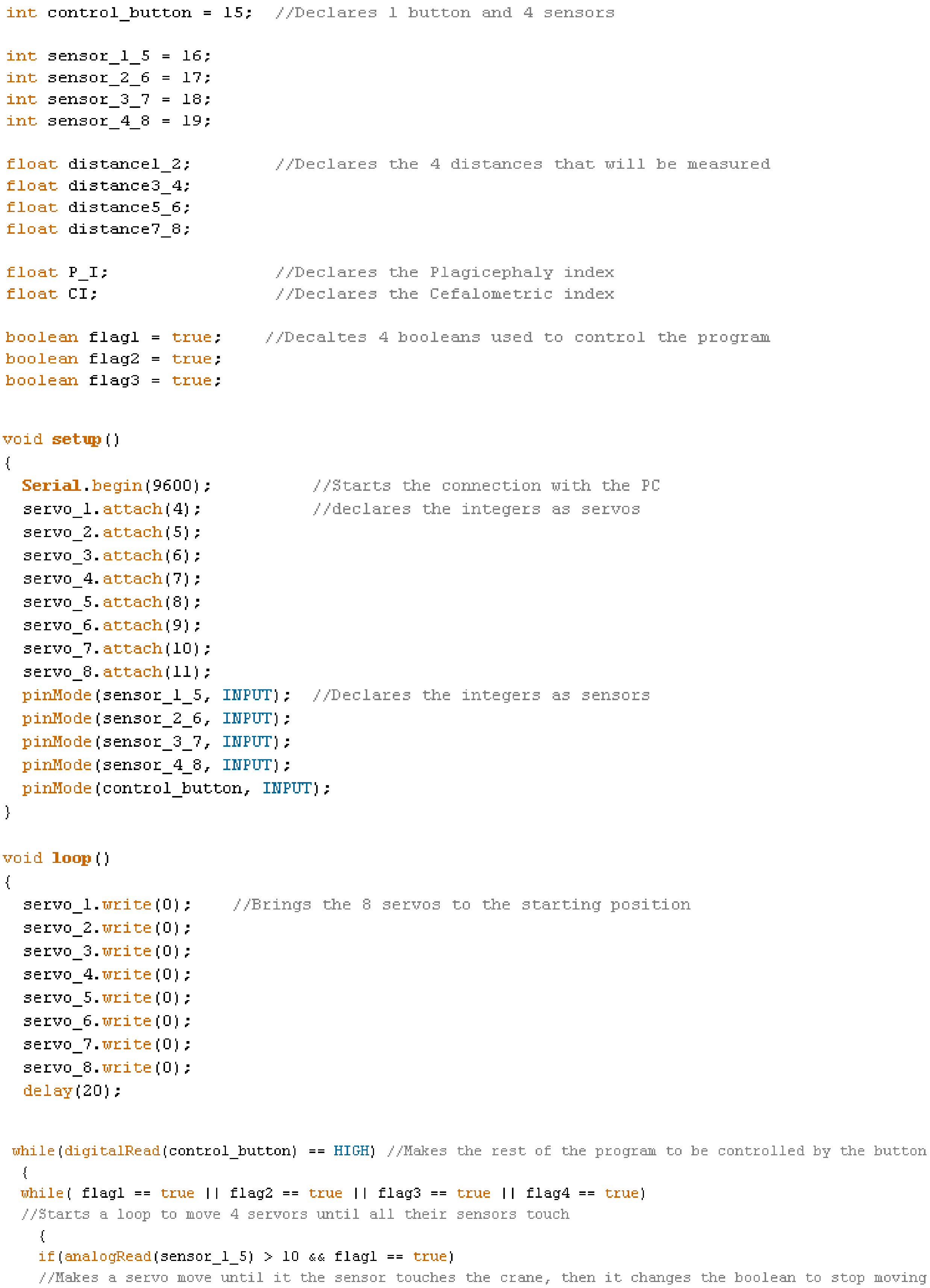
3.3.4.2.Final program

What I wanted in this program was to do a simultaneous measurement with the arms and send the correct diagnose to the computer.

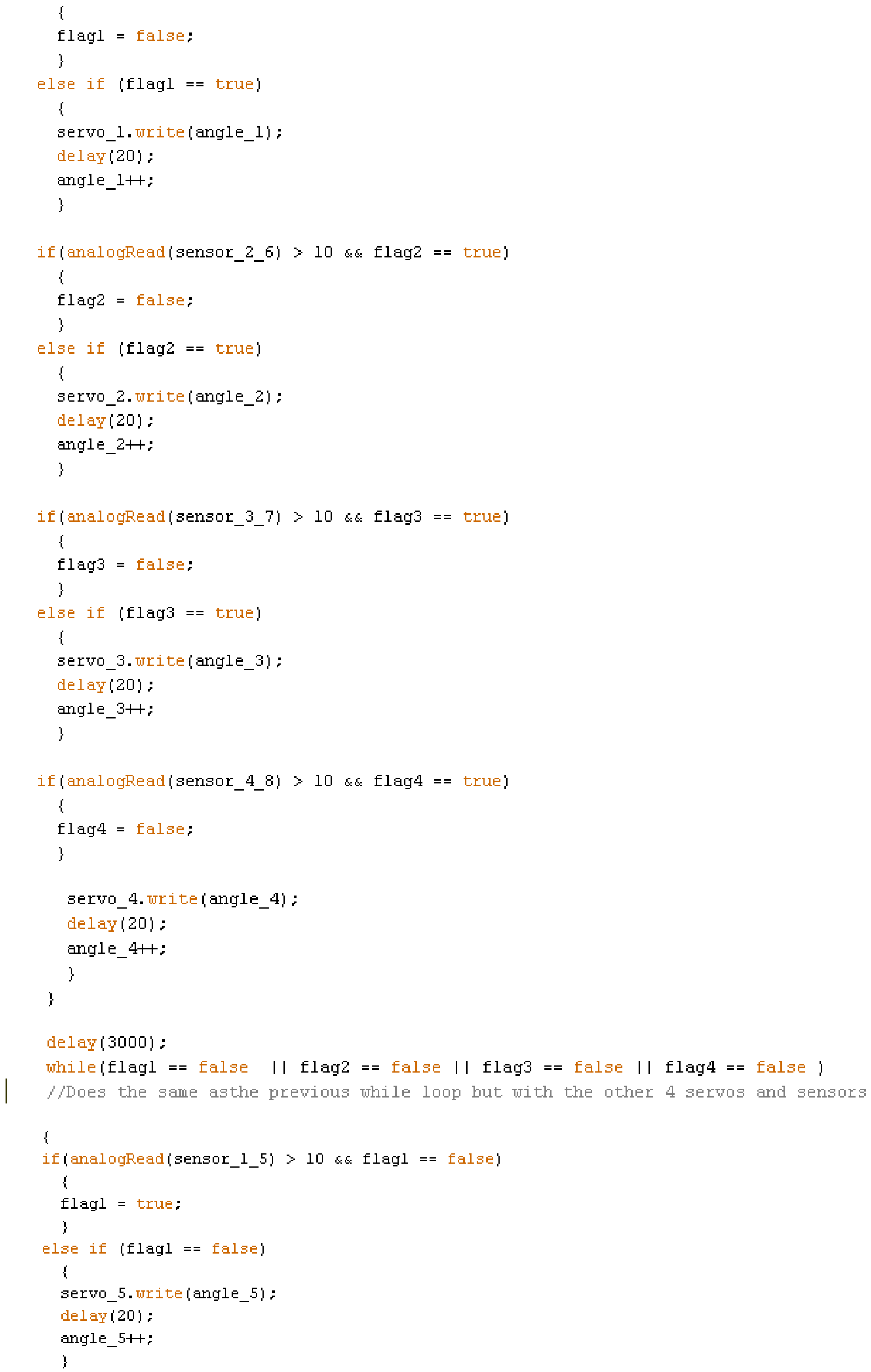
I used a new type of variable in this program called bool or Boolean, these variables can have only two values and I use as an indicator that shows if the movement of each arm has been completed.



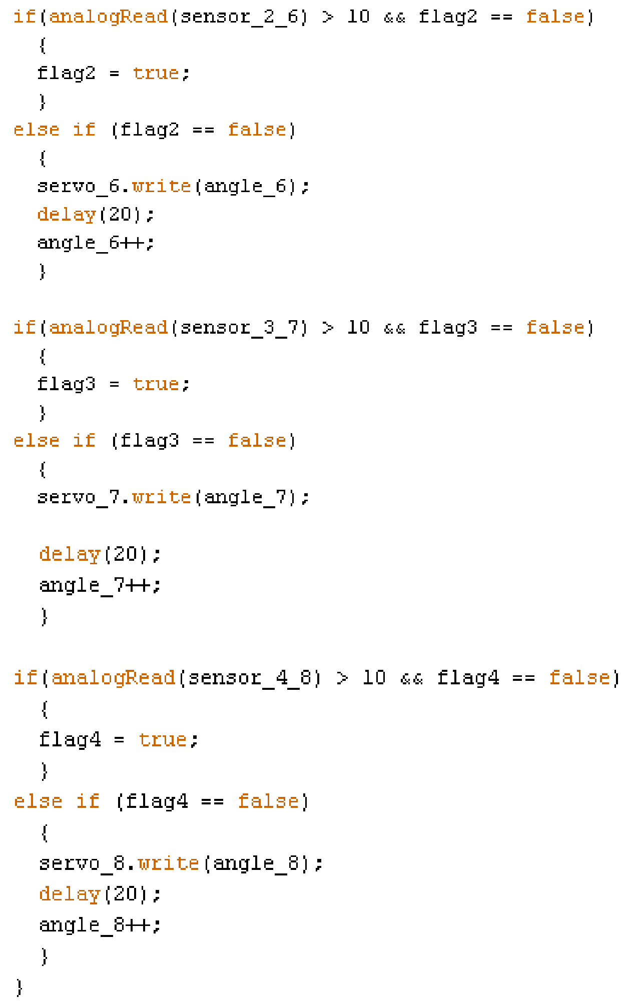
```
#include <Servo.h>  //Includes the library to use servos

Servo servo_1;      //Declares 8 servos
Servo servo_2;
Servo servo_3;
Servo servo_4;
Servo servo_5;
Servo servo_6;
Servo servo_7;
Servo servo_8;

int angle_1 = 0;    //Declares 8 angles
int angle_2 = 0;
int angle_3 = 0;
int angle_4 = 0;
int angle_5 = 0;
int angle_6 = 0;
int angle_7 = 0;
int angle_8 = 0;
```



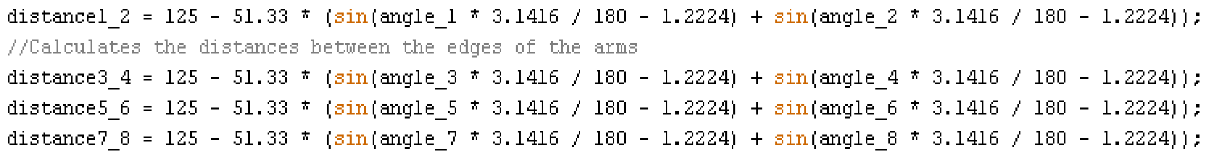
```
int control_button = 15;  //Declares 1 button and 4 sensors

int sensor_1_5 = 16;
int sensor_2_6 = 17;
int sensor_3_7 = 18;
int sensor_4_8 = 19;

float distance1_2;         //Declares the 4 distances that will be measured
float distance3_4;
float distance5_6;
float distance7_8;

float P_I;                 //Declares the Plagicephaly index
float CI;                  //Declares the Cefalometric index

boolean flag1 = true;     //Decaltes 4 booleans used to control the program
boolean flag2 = true;
boolean flag3 = true;


void setup()
{
  Serial.begin(9600);           //Starts the connection with the PC
  servo_1.attach(4);            //declares the integers as servos
  servo_2.attach(5);
  servo_3.attach(6);
  servo_4.attach(7);
  servo_5.attach(8);
  servo_6.attach(9);
  servo_7.attach(10);
  servo_8.attach(11);
  pinMode(sensor_1_5, INPUT);   //Declares the integers as sensors
  pinMode(sensor_2_6, INPUT);
  pinMode(sensor_3_7, INPUT);
  pinMode(sensor_4_8, INPUT);
  pinMode(control_button, INPUT);
}

void loop()
{
  servo_1.write(0);    //Brings the 8 servos to the starting position
  servo_2.write(0);
  servo_3.write(0);
  servo_4.write(0);
  servo_5.write(0);
  servo_6.write(0);
  servo_7.write(0);
  servo_8.write(0);
  delay(20);


 while(digitalRead(control_button) == HIGH) //Makes the rest of the program to be controlled by the button
  {
  while( flag1 == true || flag2 == true || flag3 == true || flag4 == true)
  //Starts a loop to move 4 servors until all their sensors touch
    {
    if(analogRead(sensor_1_5) > 10 && flag1 == true)
    //Makes a servo move until it the sensor touches the crane, then it changes the boolean to stop moving
```



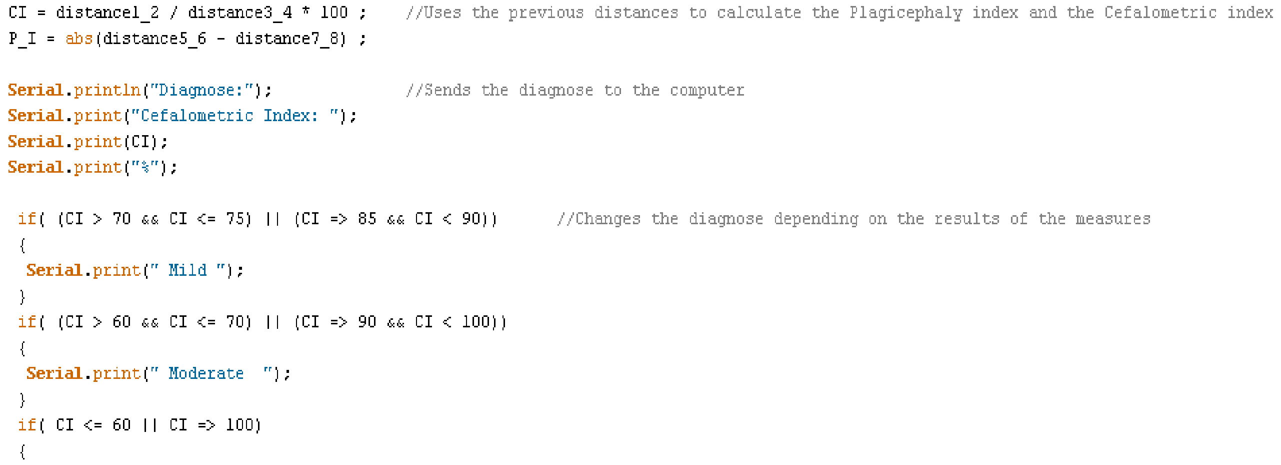
```
  {
  flag1 = false;
  }
else if (flag1 == true)
  {
  servo_1.write(angle_1);
  delay(20);
  angle_1++;
  }

if(analogRead(sensor_2_6) > 10 && flag2 == true)
  {
  flag2 = false;
  }
else if (flag2 == true)
  {
  servo_2.write(angle_2);
  delay(20);
  angle_2++;
  }

if(analogRead(sensor_3_7) > 10 && flag3 == true)
  {
  flag3 = false;
  }
else if (flag3 == true)
  {
  servo_3.write(angle_3);
  delay(20);
  angle_3++;
  }

if(analogRead(sensor_4_8) > 10 && flag4 == true)
  {
  flag4 = false;
  }

   servo_4.write(angle_4);
   delay(20);
   angle_4++;
   }
 }

 delay(3000);
 while(flag1 == false  || flag2 == false || flag3 == false || flag4 == false )
 //Does the same asthe previous while loop but with the other 4 servos and sensors

 {
 if(analogRead(sensor_1_5) > 10 && flag1 == false)
   {
   flag1 = true;
   }
 else if (flag1 == false)
   {
   servo_5.write(angle_5);
   delay(20);
   angle_5++;
   }
```



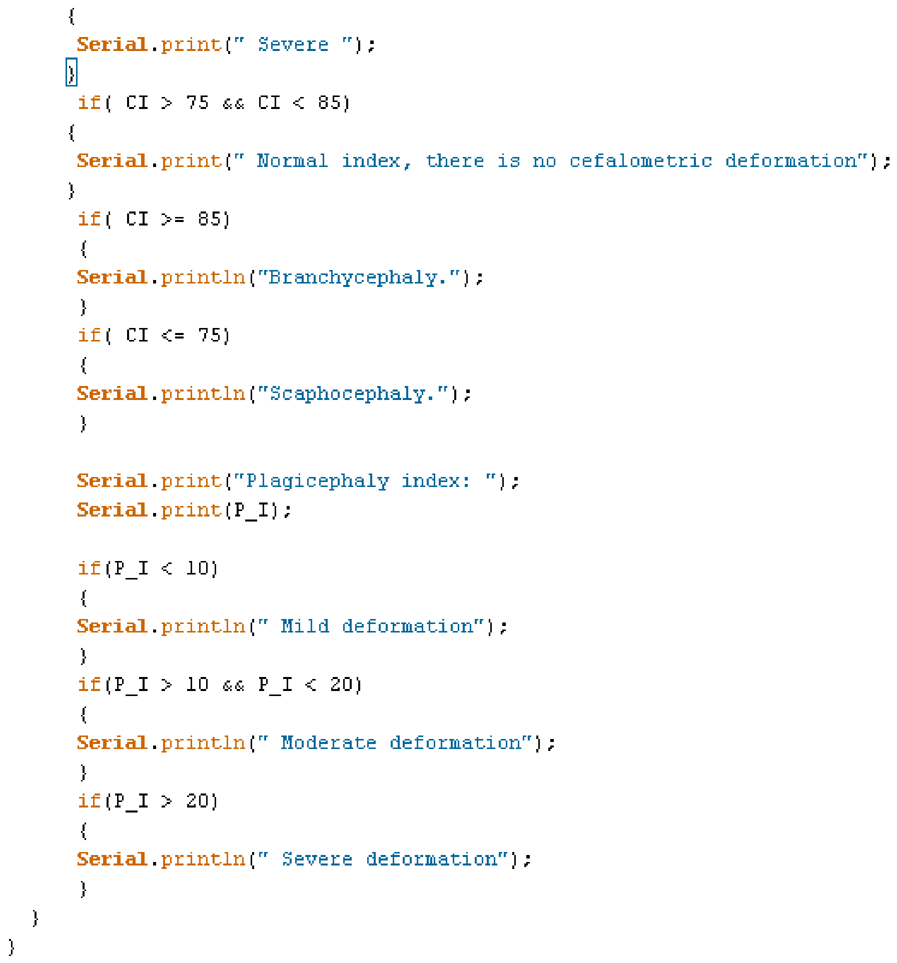
```
if(analogRead(sensor_2_6) > 10 && flag2 == false)
  {
  flag2 = true;
  }
else if (flag2 == false)
  {
  servo_6.write(angle_6);
  delay(20);
  angle_6++;
  }

if(analogRead(sensor_3_7) > 10 && flag3 == false)
  {
  flag3 = true;
  }
else if (flag3 == false)
  {
  servo_7.write(angle_7);

  delay(20);
  angle_7++;
  }

if(analogRead(sensor_4_8) > 10 && flag4 == false)
  {
  flag4 = true;
  }
else if (flag4 == false)
  {
  servo_8.write(angle_8);
  delay(20);
  angle_8++;
  }
}
```



```
distance1_2 = 125 - 51.33 * (sin(angle_1 * 3.1416 / 180 - 1.2224) + sin(angle_2 * 3.1416 / 180 - 1.2224));
//Calculates the distances between the edges of the arms
distance3_4 = 125 - 51.33 * (sin(angle_3 * 3.1416 / 180 - 1.2224) + sin(angle_4 * 3.1416 / 180 - 1.2224));
distance5_6 = 125 - 51.33 * (sin(angle_5 * 3.1416 / 180 - 1.2224) + sin(angle_6 * 3.1416 / 180 - 1.2224));
distance7_8 = 125 - 51.33 * (sin(angle_7 * 3.1416 / 180 - 1.2224) + sin(angle_8 * 3.1416 / 180 - 1.2224));
```



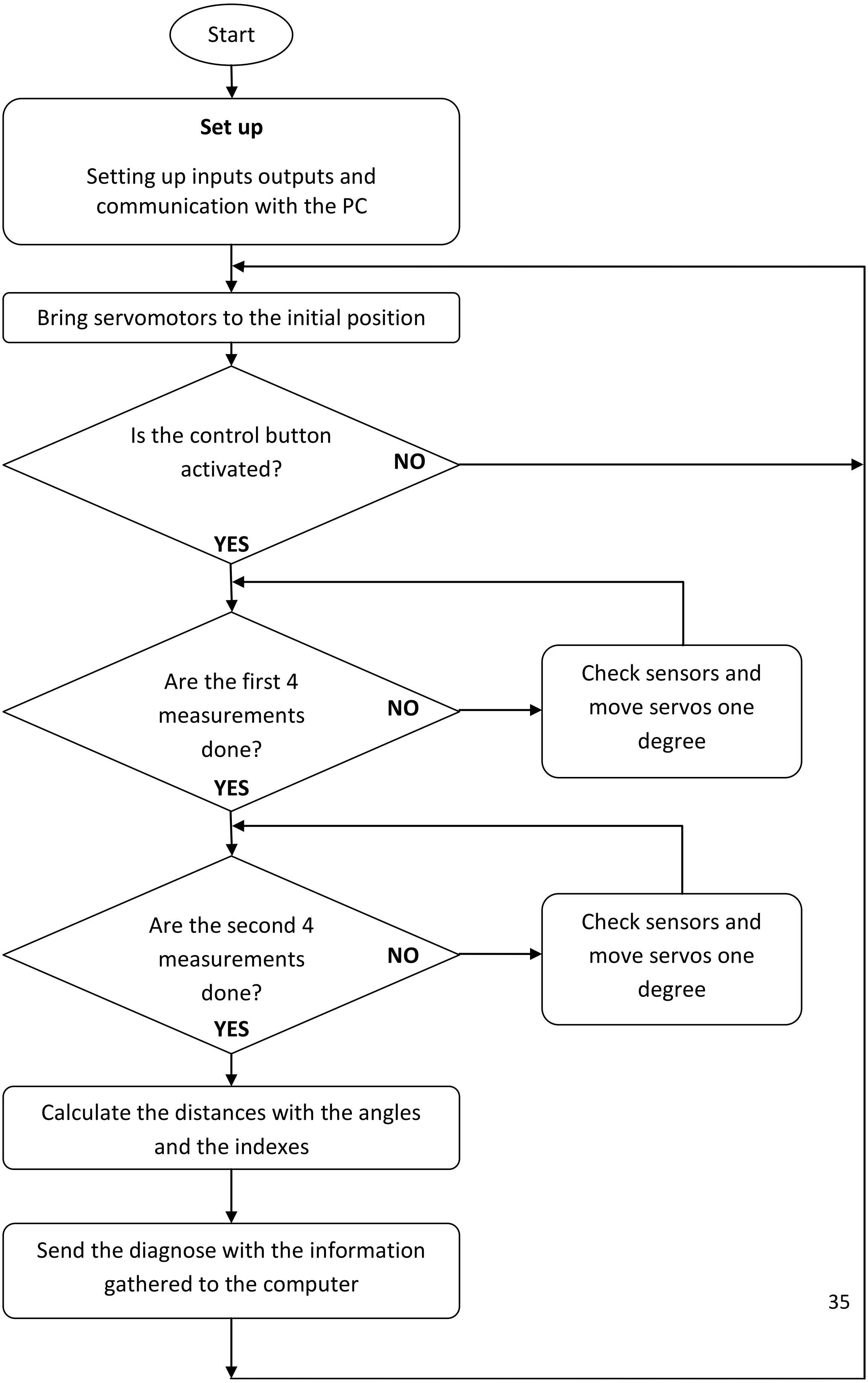
```
CI = distance1_2 / distance3_4 * 100 ;    //Uses the previous distances to calculate the Plagicephaly index and the Cefalometric index
P_I = abs(distance5_6 - distance7_8) ;

Serial.println("Diagnose:");              //Sends the diagnose to the computer
Serial.print("Cefalometric Index: ");
Serial.print(CI);
Serial.print("%");

 if( (CI > 70 && CI <= 75) || (CI => 85 && CI < 90))      //Changes the diagnose depending on the results of the measures
 {
  Serial.print(" Mild ");
 }
 if( (CI > 60 && CI <= 70) || (CI => 90 && CI < 100))
 {
  Serial.print(" Moderate  ");
 }
 if( CI <= 60 || CI => 100)
 {
```



```
     {
      Serial.print(" Severe ");
     }
      if( CI > 75 && CI < 85)
     {
      Serial.print(" Normal index, there is no cefalometric deformation");
     }
      if( CI >= 85)
      {
      Serial.println("Branchycephaly.");
      }
      if( CI <= 75)
      {
      Serial.println("Scaphocephaly.");
      }

      Serial.print("Plagicephaly index: ");
      Serial.print(P_I);

      if(P_I < 10)
      {
      Serial.println(" Mild deformation");
      }
      if(P_I > 10 && P_I < 20)
      {
      Serial.println(" Moderate deformation");
      }
      if(P_I > 20)
      {
      Serial.println(" Severe deformation");
      }
  }
}
```

3.3.4.3.Final program algorithm

This algorithm shows the step-by-step logical procedure of the program, a set of rules that precisely defines a sequence of operations.


Start
Set up
Setting up inputs outputs and communication with the PC
Bring servomotors to the initial position
Is the control button activated?
NO
YES
Are the first 4 measurements done?
NO
YES
Check sensors and move servos one degree
Are the second 4 measurements done?
NO
YES
Check sensors and move servos one degree
Calculate the distances with the angles and the indexes
Send the diagnose with the information gathered to the computer

3.4.A huge problem

Although the prototype was thought to have 8 mobile arms with 8 sensors and 8 servos, I had to reduce the amount of arms due to malfunctions and breakdowns of some servos and an unfortunate loss of a sensor. Therefore, I removed one of the axes making the prototype unable to perform a reliable diagnosis of the cefalometric index.

Besides, I found some problems on the performance of the control button, consequently I decided to put it aside and use a detection of a sensor as a trigger to start the program.

Doing such big changes meant a change of the final program which was adapted to a partial diagnose[2].

4.Marketing

In the world we are living all the new ideas are commercialized as soon as possible, this is the reason why I bore in mind a commercialization of the tool when doing the project. Therefore, this idea has some features that favour its release to the market as a product:

-Solid and strong structure: Offers handiness and eases its use.

-Integrated program: Loading only one program to the microcontroller makes a mass production possible.

-Standard USB communication with a computer: Ensures a wide range of compatibility.

-Relative low cost: With a prototype cheaper than 150 €, its price would fall with a mass production.

-Use of open-source technology: Avoids paying fees for the usage of technology.

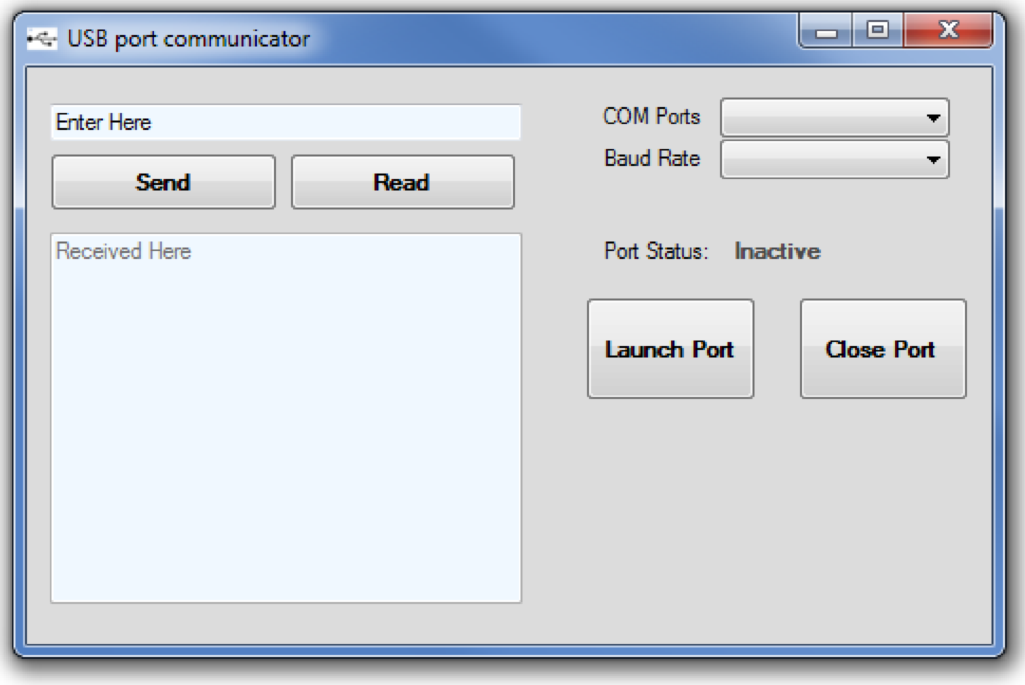
[2] The code of the real final program can be found in the third annex.

-Long-term utility: The tool is useful until something improves the detection of cranial deformations.

Despite these advantages, I found a huge barrier to commercialisation; to have the message printed on screen I used the Arduino development environment, to do that I had to install this program on my computer. As I do not want the costumers to spend too much time setting up their computer to use the product, I searched an easier way to show the message on screen.

The solution is a program for computer that shows the communication with a serial port (USB port)[3]. This program is in C++ as it is an extended programming language and it is easy to run in any Windows™ operative system. As doing this program was not the aim of my project I asked for help to two friends that were also working with C++ and we adapted an existing program to suit my purposes.

Appearance of the program:



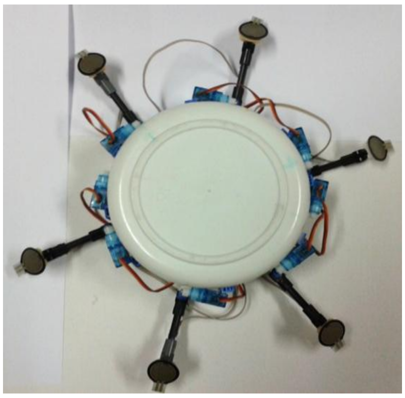
Port Status: Inactive

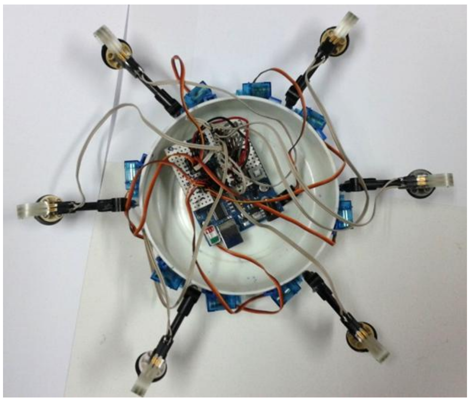
[3] The code of the C++ program can be found in the first annex and the program is attached to the research project.

5.Results

At this point of the project, I had already done most of the work so it was time to evaluate how it works.

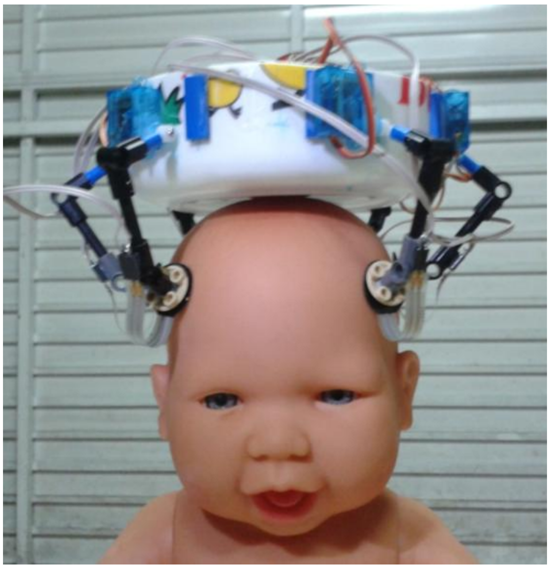
I have a structure with 6 mobile arms, I have six servomotors and sensors ready to use and a microcontroller with a program loaded. This program is supposed to do a simultaneous measure of two axes and then the other one. Then, a message with the diagnose is supposed to appear on screen[4].





After more than 25 tests, I concluded that the electronic tool is able to measure distance with an accuracy of ± 2mm in a range from 90mm to 150mm, however I encountered a phenomenon that was a huge drawback. I named it the "shaking"[5], it occurs when the device has to do multiple measures and it supposes a bad performance of a servo has at going to a determinate position. Consequently, some of the measurements lose some accuracy.

I put lots of effort trying to fix the "shaking" by changing the program many times in many ways, however, I did not succeeded, I could only minimize its appearance by slowing the work of the program and the servos. Otherwise, I came to the resolution that the cause of this phenomenon is either in the microcontroller's performance, in the servomotors' performance or in the electrical connection's condition.



[4] A video of the prototype working is attached to the research project.

[5] A video of the "shaking" phenomenon is attached to the research project.

6.Conclusion

Learning about cranial deformations and how to deal with them has enabled me to synthesise the needs of this context in a solution.

I have ideated a tool that would improve the diagnose made in primary attention by providing an objective examination. I have achieved success at creating a strong and handy structure, however, the programming and electronics have become a handicap as I have found problems preventing me from achieving the model prototype.

Moreover, the toughness of this challenge encouraged me and made me aim for the best outcome. Despite not succeeding completely, I feel I have invested effort and I have got several results, therefore I am satisfied.

Eventually, this research project has taught me several things, probably more than I actually expected; I have learnt: to do a deep research, lots of information about cranial deformations, a new programming language, how to do a formal document, that electronics are difficult to deal with, to do digital technical drawing...

And I have learnt that:

A good work is a process where failures' learning leads to success.

7.Bibliography

Documents:

Joan Bosch i Hugas, Josep Maria Costa i Clara: *La plagiocefàlia posicional: una tasca d'Atenció Primària*, Generalitat de Catalunya and Sant Joan de Déu, 1st edition, January 2012.

Audiovisual recources:

Plagiocefàlia posicional: pautes de diagnòstic, prevenció, seguiment i derivació des de l'Atenció Primària. Hospital Sant Joan de Déu BCN, [Conference], Available online at <http://vimeo.com/30720927>, 56:50 min.

Websites:

Pololu Robotics and Electronics [on line][Checked on January 2013]Available on the net.

<http://www.pololu.com/>

Arduino - HomePage [on line][Checked on January 2013]Available on the net.

<http://www.arduino.cc/>

Wikipedia, the free encyclopedia [on line][Checked on January 2013]Available on the net.

<http://en.wikipedia.org/>

Prevención y tratamiento de la plagiocefalia y la branquicefalia [on line][Checked on January 2013]Available on the net.

<http://www.infocefalia.com>

8.Acknowledgements

A research project is a time-consuming task that demands will and dedication, for this reason I would like to thank my family for their support and their disposition to help.

Secondly, I am grateful to my tutor Adoración Medrano whose aid has guided me in the search of a great research project.

In addition, I thank Miquel Navarro and Victor Ferrer that helped me with the programming in C++.

Moreover; I acknowledge the work of the Arduino project as it provides a great basis of electronics available for everyone.

Eventually, I want to show gratitude to Pilar Gussinyé who has contributed in this project bringing me a close approach to the professional environment where this project is set.