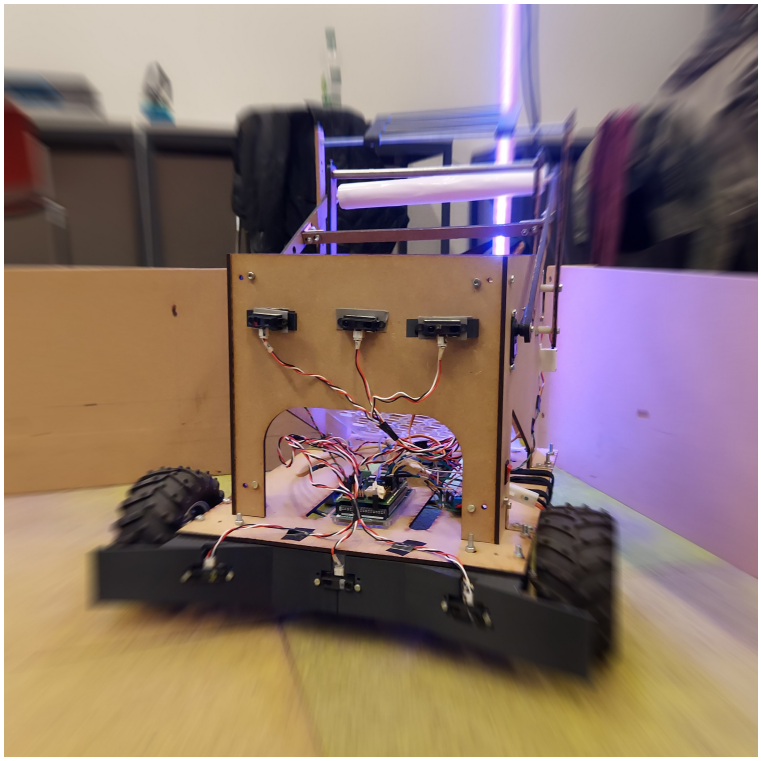
# STI Robot competition
# Team 3 - *GaliPET*

Martin CIBILS
Maxime ROMBACH
Nils TOGGWYLER



Supervisor: Alessandro Crespi
Coach : Özdemir Can Kara
Professor : Auke Ijspeert

January 2021

# Contents

## Acknowledgments

This following report is the result of the great adventure we had during this semester. We went through this robot competition with the help of many people who we would like thank.
We want to thank first its supervisor, Alessandro Crespi, who was always available for help and advice when we needed. With Prof. Auke Ijspeert, they are the people who keep this competition from year to year. This allows more and more students to live this formidable experience. The responsible of SKIL, Stéphane Clerc and Marc Wettstein, were also full of resources when we had trouble with MEA 330 laser cutter. Our coach Özdemir Can Kara who followed us during the process and gave us advice.

# 1 Introduction

This report deals with our solution concerning the interdisciplinary project *Robot Competition*. Its objective is to build a robot from scratch capable of autonomously picking up PET bottles and bring them back a the recycling zone of an arena. The later is illustrated in Fig.1

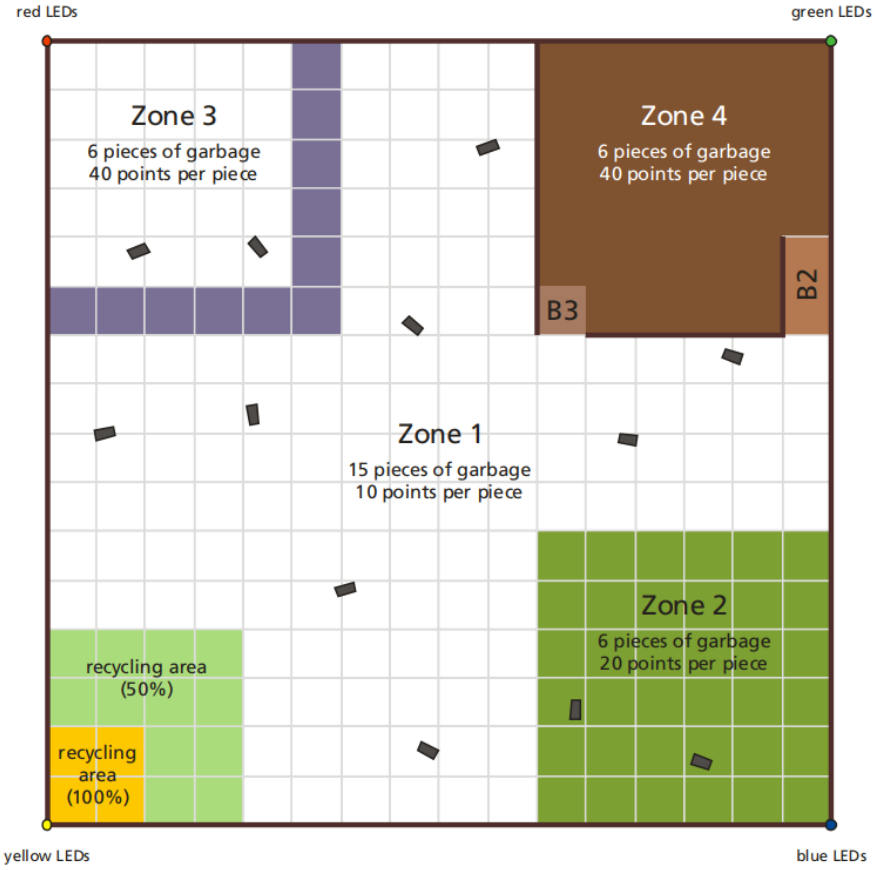



Figure 1: Illustration of the competition arena

The 8×8m arena is made of 0.5x0.5m carpet tiles and is divided in 4 zones in which the bottles are worth different points:

- Zone 1: regular flat ground carpet
- Zone 2: synthetic grass
- Zone 3: regular carpet surrounded by rocks
- Zone 4: elevated platform with a slope

In addition of this global mapping, several cinder blocks are added randomly as local obstacles.

# 2 Functional analysis

As none of us had much record in robot building before, this competition was a great learning challenge. That is why we quickly decided to look for a design as simple as possible. This section summarizes the ideas we had at the beginning with their pros & cons.

## 2.1 Locomotion

This part is crucial for the robot to be able to move properly. We summarized what we thought being the pros & cons of each method.

Table 1: Pros & Cons for **locomotion**

| | Pros | Cons |
|---|---|---|
| 2 driving wheels | • Simple to control (only 2 motors)<br>• Odometry | • Need powerful motors to climb rocks or slope |
| 4 driving wheels | • Less need for powerful motors<br>• Easier to pass the rocks as the 4 wheels participate to the motion | • Not easy to control/steer, compared to 2 wheels solution<br>• Difficult odometry |

We chose the 2 driving wheels because our final strategy did not involve the rocky or elevated zones and the additional power was not needed.

## 2.2 Vision

This section deals with the matter of the robot's vision. It differentiates 2 sides: obstacle & bottle detection, respectively repelling and attracting the robot. The functional analysis are similar in both case with one difference: Information about distance is quite important when it comes to obstacle avoidance, though it is not to be neglected for bottle detection.

Table 2: Pros & Cons for **bottle detection** methods

| | Pros | Cons |
|---|---|---|
| Camera with ML model | • Model pretty accurate once trained<br>• Reliable<br>• Differentiate easily bottle/obstacle | • Need computational power, more electronics parts<br>• Takes time to train if unfamiliar with it |
| Proximity sensors (IR, ultrasonic) | • Easy to implement<br>• No need for high computational power | • Less reliable for bottle detection<br>• Need special arrangement to efficiently detect bottles and differentiate obstacles |

Our first choice was the camera because we thought the precision was worth the trouble. However, the results did not follow so we finally fell back on the IR sensors option.

Table 3: Pros & Cons for **obstacle avoidance** methods

| | Pros | Cons |
|---|---|---|
| Camera with Machine Leaning model | • Model pretty accurate once trained<br>• Reliable<br>• Differentiate easily bottle/obstacle | • Need computational power, more electronics parts<br>• Takes time to train if unfamiliar with it<br>• **Feasible but difficult to obtain distance measurement** |
| Proximity sensors (IR, ultrasonic) | • Easy to implement<br>• No need for high computational power<br>• **Designed for distance measurement** | • Less reliable for obstacle detection<br>• Need special arrangement to efficiently detect bottles and differentiate obstacles |

For the obstacle, high range detection was never really needed so the obvious choice was the IR sensors.

## 2.3 Bottle displacement

This feature concerns the way of moving and storing the bottles to bring them to the recycling area.

Table 4: Pros & Cons for **bottle displacement** methods

| | Pros | Cons |
|---|---|---|
| Claw | • No difference between lying & standing bottle<br>• Reliable once implemented | • Need many actuators to work |
| Sticky arm | • One actuator is enough<br>• Simple | • Can become unreliable<br>• Need to unstick bottles<br>• Need to be precise<br>• Standing bottles |
| No grab, bottle under robot | • Simplest<br>• No need for actuator | • Rocks impossible<br>• Precision less crucial |
| Conveyor belt | • One actuator | • Precision less crucial |

The sticky arm was chosen because it seemed like the most reliable option and also because it had the simplest parts.

Table 5: Pros & Cons for **bottle storage** methods

| | Pros | Cons |
|---|---|---|
| Container | • Can store many bottles | • Heavier & bulkier robot |
| No container | • Light robot | • Limited number of bottles |

Having a container was more suited to our strategy it terms of trip efficiency.

## 2.4 Localization & Navigation strategy

The robot has to be able to localize itself in order to be able to implement a searching strategy.

Table 6: Pros & Cons for **localization** methods

| | Pros | Cons |
|---|---|---|
| Odometry with encoders | • Relatively easy to implement | • If wheels slip, position lost |
| Compass | • Absolute measure of angle<br>• Easy to retrieve value | • Very sensitive<br>• Needs to be placed high over the ground |
| 360° vision | • Absolute meaurement of position & angle | • Requires high computational power<br>• Sensitive to ambiant light |
| IMU | • Measurement of many coordinates with one sensor<br>• Resistant against kidnapping | • Drifts greatly |
| LIDAR | • Provides an accurate map of the arena with obstacles<br>• Very powerful | • Need high computational power<br>• Not so easy to implement<br>• Bulky sensor |

Our selected strategy involved an odometry based localization that was recalibrated by a compass because it was the most attainable in terms of implementation.

Table 7: Pos & Cons for **search strategies**

| | Pros | Cons |
|---|---|---|
| Random search only | • Relatively easy to implement | • Not very efficient, outcome higly depends on initial bottle placement |
| Go in a zone & random search in it | • Reduces random search to a smaller area | • Need to have a good localization to be sure to reach the wanted zone |
| Go in a zone & grid search in it | • Quite efficient to cover a whole area | • Need good localization and control for trajectory tracking |

Considering our localisation choices, the random search was the most suited to our needs because of its relatively low precision. The random search is robust in the sense that it can still work even with large errors.

# 3 Experiments

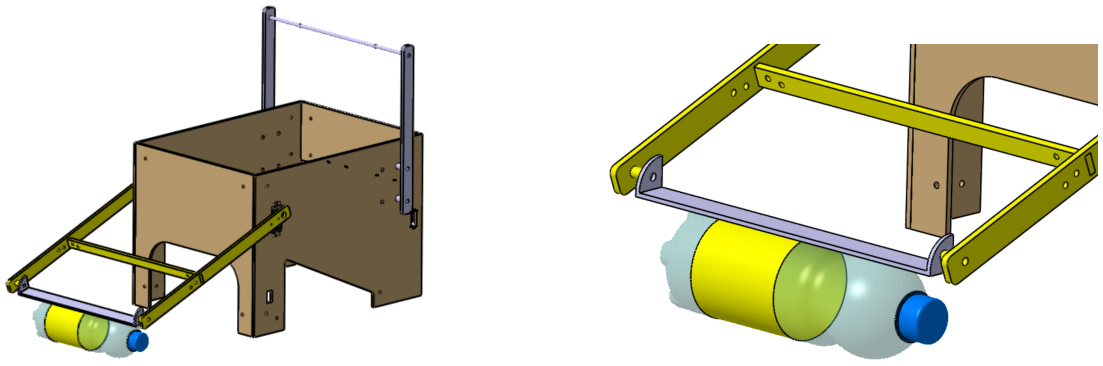
During our design process, we did some experiments to confirm or reject some of the principal ideas. The first one being our solution for moving the bottles.

## 3.1 Bottle grabbing

As we wanted to have a simple and functional design to our standards, we quickly had the idea of a sticky arm. The basic idea is to have a rotating arm after which a bar with double sided tape underneath.
Since this method can easily become unreliable depending on its implementation (arm design, choice of tape etc), our coach strongly advised us to extensively test our idea to be sure that it is able to attach and detach a bottle properly.

### Sticking the bottle

We then did tests for the part where the tape was attached. This way we wanted to ensure that the adhesive properties worked well.
First tried with 1 stop and a tapped bar to have a large sticky area (Fig.2).

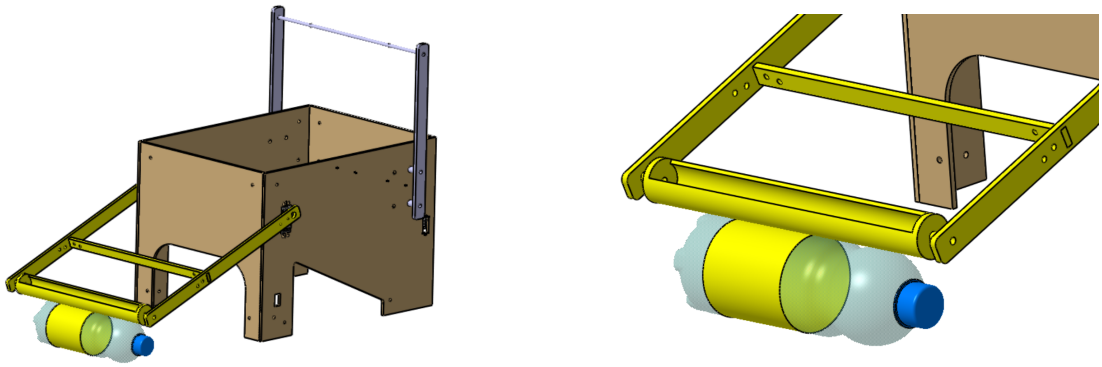


(a) Isometric view (b) Zoom on the bar

Figure 2: Very first prototype idea: 1 flat bar & 1 stop

This method has one major inconvenient: the efficiency of the flat bar highly dependent of the angle at which it is oriented. Meaning that if it is not parallel to the ground when it touches the bottle, it will not

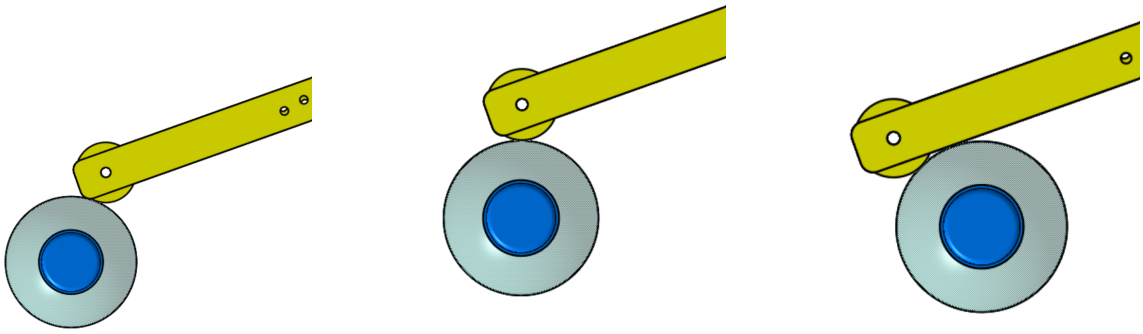
attach properly. Plus, if the robot is not precisely placed in front of the bottle, it will not stick either. Facing that problem we changed the shape of the flat bar for a tube. (Fig. 3)



(a) Isometric view (b) Zoom on the tube

Figure 3: 1 tube & 1 stop

With a tube the tolerance on the placement of the bottle with respect to the robot is much larger (Fig.4). However, the drawback of this solution is a reduction of the sticking area. Since we noticed that we had a quite strong double sided tape, we decided that it compensate this reduction.

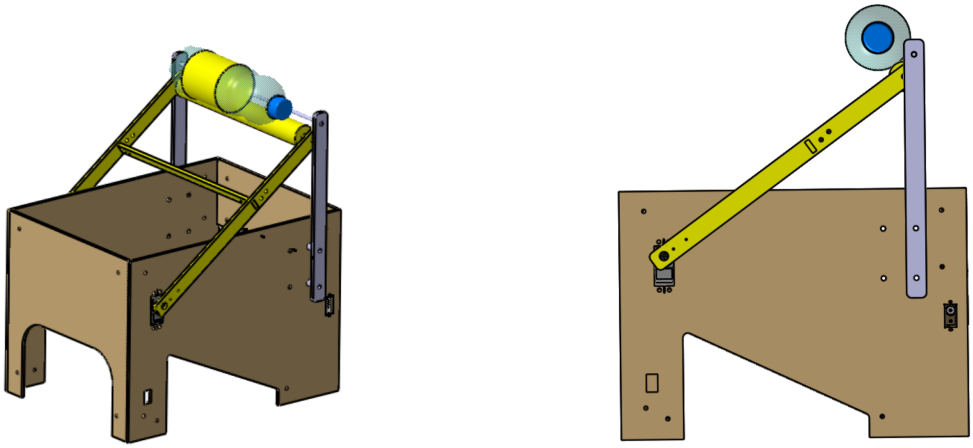


(a) Bottle slightly in front (b) Perfect placement (c) Bottle slightly at the back

Figure 4: Bottle placement tolerance with taped tube

**Unsticking the bottle**

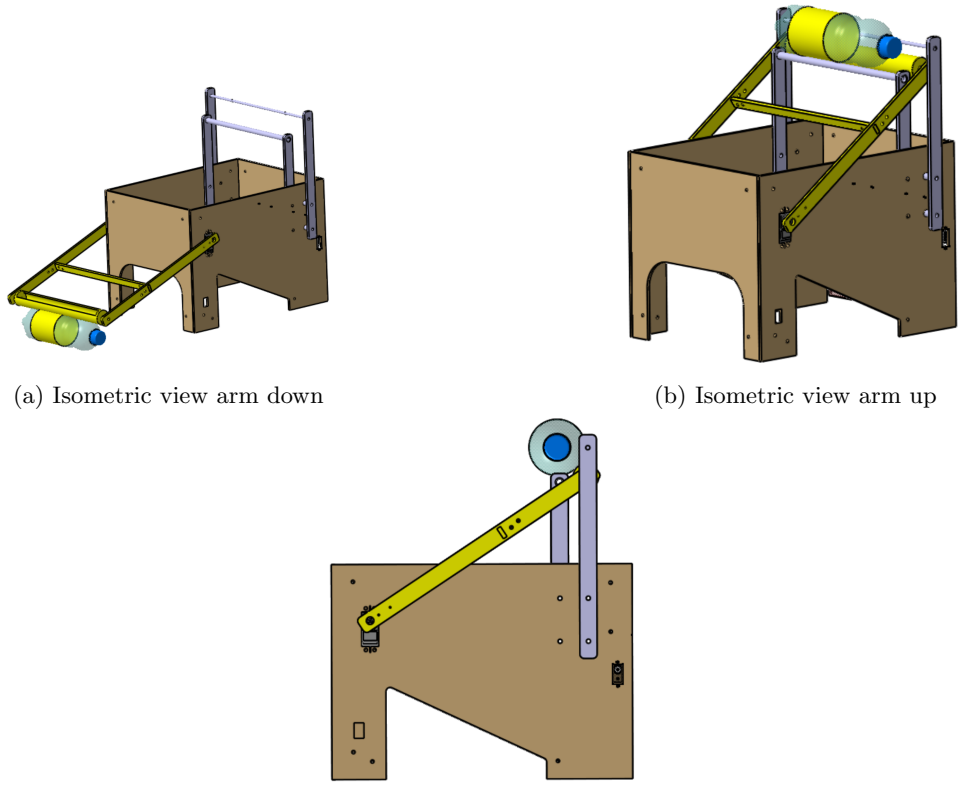
Now that we had a satisfying sticking strategy, we had to think about how to unstick the bottle. We first thought about a single stop on which the bottle would contact while the arm continues to rotate. This should cause the bottle to detach.



(a) Isometric view (b) Right view

Figure 5: 1 tube & 1 stop with contact bottle

However, this solution revealed to be unefficient as the bottle simply stayed stuck on the tape while rolling on the tube when the contact occurred. Then, we decided to add another stop to prevent this phenomenon (Fig. 6).



(a) Isometric view arm down (b) Isometric view arm up

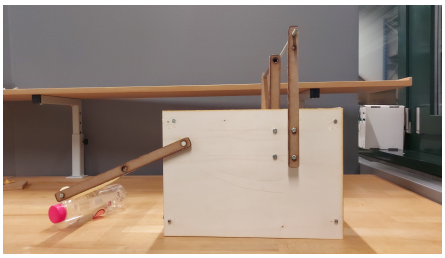
(c) Right view arm up

Figure 6: 1 tube & 2 stops with contact bottle

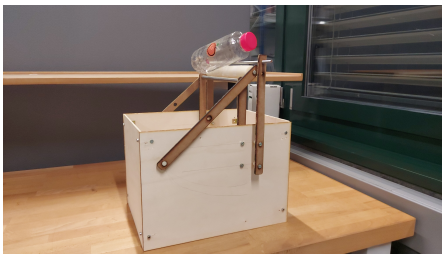
The prototyping (Fig.7) was done with laser cut MDF, spare wood and spare metallic bar that we found

in the lab. In turned out to be quite convincing[1] so we decided to keep this mechanism.

(a) Overview: 1 rotating sticky arm & 2 stops



(b) Attaching the bottle

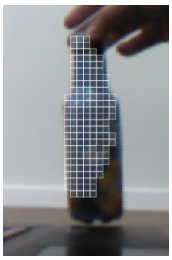


(c) Detaching the bottle against stops

Figure 7: First working prototype for bottle grab

## 3.2 Pixy

One of our ideas and experiment was to use a Pixy Camera to detect the bottles. Indeed the pixy camera works pretty well on color full objects. The Pixy camera works in a way such that you train it to detect objects with a specific color. Since the bottles had some colors in them we had hope that with some training it would be able to detect them. After some testing, the results were not convincing enough as some bottle with a uniform color were well detected but other with different or not enough color had a bad result as it can be seen on fig 8. In the end we decided not to use the Pixy camera for out robot.



(a) Pixy with differents colors on bottle

(b) Pixy with not enough color on bottle

(c) Pixy on uniform color bottle

Figure 8: Example of Pixy Camera on bottles

## 3.3 Compass

One of our method for localization consisted in using odometry. Using encoders on the wheels we knew that it could provide a pretty acceptable estimation of the localization. But we also knew that this technique

[1]For proof of concept: video on lying bottle & standing bottle

would drift with time. The coordinate that was the most likely to do so was the angle of the robot (Fig. 9)

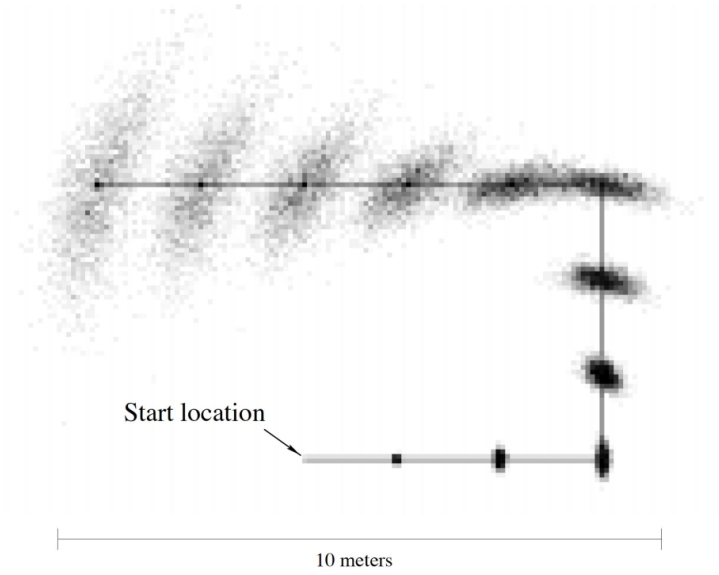



Figure 9: Illustration of odometry drift: it increases less when the robot moves straight than when it turns [1]

That is why we wanted to recalibrate the angle with an absolute sensor such as a compass. We used the *GY-26* provided in the catalog (Fig.10).

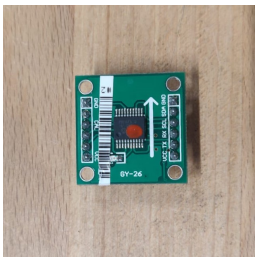


Figure 10: Compass *GY-26* as provided from the catalog

M. Crespi told us that the compass was a very sensitive sensor. Meaning which, it had to be carefully calibrated and placed high above the ground in order to avoid as much noise as possible. That is why we decided to build an antenna after which we fixed the compass. Its resulted in height of approximately 75 cm above the ground (Fig.11). We noticed a slight improvement of the signal compared to when it was placed closer to the ground.

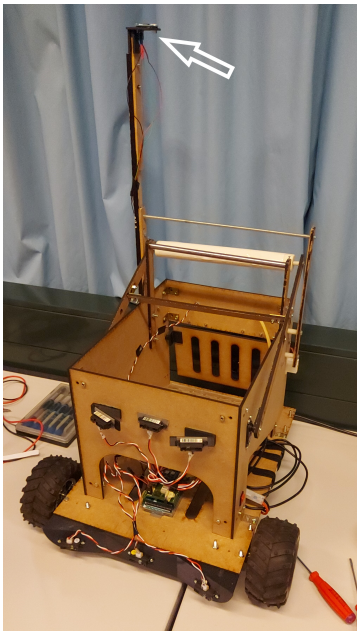


Figure 11: GaliPET with antenna mounted. The white arrow indicates where the compass is placed

Unfortunately we still had many issues with this sensors.
The first one came from the arm servo MG995R. Indeed, when it was under voltage it caused important noise to the compass' signal, which became useless.
In normal use delivered a value with a tolerance of 1° which was acceptable. However, when the servo was powered (red arrow in Fig.12), it became erratic[2].

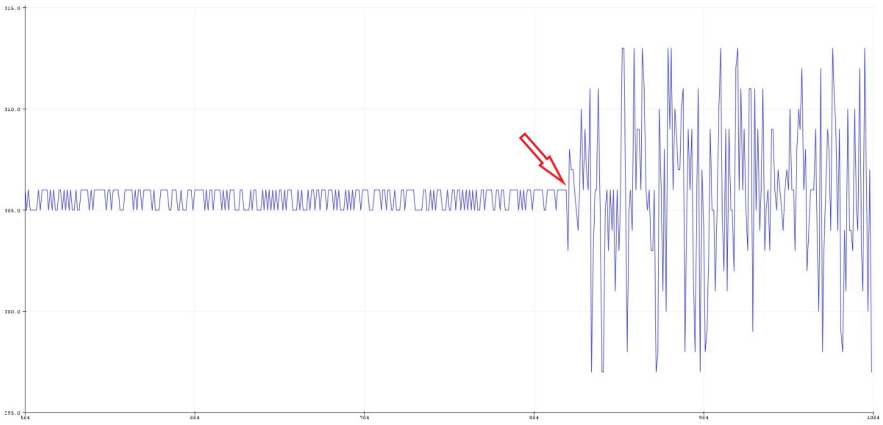


Figure 12: Behavior of the compass' signal before and after putting the arm servo in tension

We were able to fix this problem using the *detach()* command in our code, which allowed us to detach the arm servo's pin, thus disabling its power source while the robot did not need it.

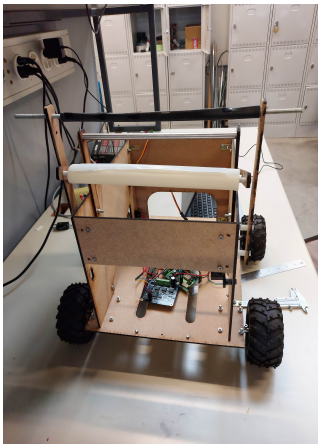
The biggest issue we had was the one that put an end to our use of compass. For a reason we still cannot explain, the angle was decreasing greatly when the robot was driving straight, while it should have remained constant and only change during turns. Our most feasible assumption was that the magnetic field in the arena was inhomogeneous. We tried to fix this problem with many calibrations process all around the arena,

[2]This comes from the serial plot of the running program on Arduino. Therefore, we were not able to retrieve the data other than through a screenshot

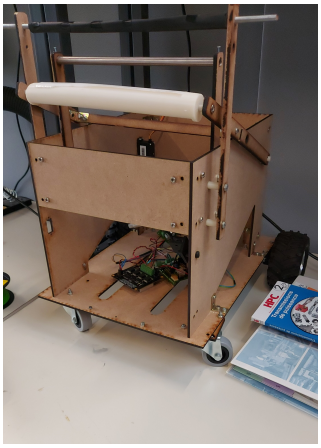
without success. These tries lasted until 2 days before the competition, until which we decided to only use odometry with encoders. We were aware that was not a perfect solution, thus we implemented a recalibration procedure detailed in Section 4.2.6

## 3.4 Wheels & Motorization

We first thought about 4 driving wheels in order to be able to pass the rocks. But at a certain point we decided that it would be simpler to use a simple differential robot with 2 driving front wheels. As 4 wheels were already placed (Fig.13a) we tried to test our differential like this. But as Prof. Ijspeert, said during the milestone presentation the resulting friction when turning turned out to be enormous, causing our robot's back to slide on the ground.
At this moment, we decided to trade our bulky back wheels for 2 back castors. This would turn our robot into a pure differential robot much easier to steer (Fig. 13b).



(a) 4 wheels version of GaliPET, the bulky back wheels produce great friction when turning



(b) 2 bulky front wheels & 2 castors version of GaliPET, turning is now much better with free wheels

Figure 13: Evolution of back wheels

# 4 Our final solution

This section will present and justify our design choices when it comes about the hardware & software. One can see the result of this implementation in the presentation video.

## 4.1 Hardware

### 4.1.1 Power source

We wanted our solution to be as simple as possible, so it did not seem that we would include many power demanding components. Thus, it quickly came to us that we would not need a powerful power source. That is why went for a single NiMH battery (7,2 V, 300 mAh). We took 2 of them so that when one of them is always fully charge. This way we did not have to wait for our battery to charge to resume our tests. The battery was directly connected to a fuseboard and a switch, allowing us to quickly power off the robot.

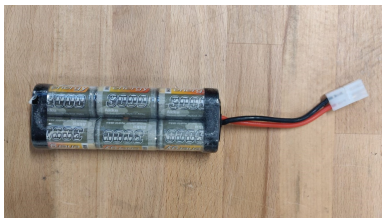
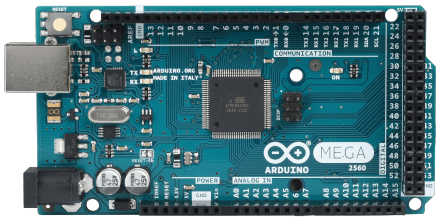


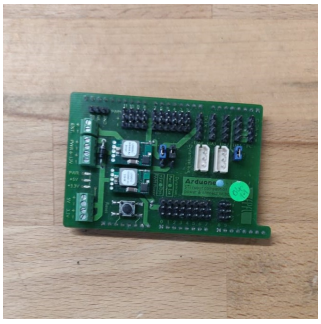
Figure 14: NiMH battery (7,2 V, 3000 mAh)

### 4.1.2 Controller

Our solution does not require a huge computational power. No image processing or heavy calculations were implemented, so in the end, there was no need for something as powerful as a Raspberry Pi for exemple. We chose GaliPET's neural center as an Arduino MEGA 2560 on which an Arduone shield is mounted to facilitate connections (Fig.15).




(a) Arduino Mega 2560



(b) Arduone shield

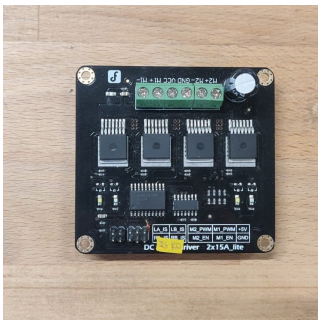
Figure 15: Elements for controller

### 4.1.3 Locomotion

#### Motorization

Regarding the locomotion we stayed on something very basic. As we chose to use NiMH batteries our choices were limited when it came down to choosing the motors. Thus we went for two 75:1 Pololu (HP) DC motor with encoders, one for each front wheel. Using an other battery such as the LiPo one could have allowed us to pick others motors such as the Maxon, more powerful and reliable. However, they were not chosen because they were not equipped with encoders, useful if one wants to implement odometry. The later were already attached to the Pololu, which was really convenient. In order to control them we had to use a driver, we went with a DFRobot DRI0018 2x15A motor driver, which had two H bridge and dual, so one driver was enough for both motors.

(a) Pololu 75:1 DC motor with encoder (HP)



(b) DFRobot DRI0018 2x15A motor driver

Figure 16: Elements for motorization

**Wheels**

Since we picked Pololu motors, the most convenient front wheels to use along were the WildTumper 120×60 mm (Fig. 17) from the catalog. Indeed, they had an inbuilt shaft adapter for those motors which made them quite easy to attach.
Concerning the back, we decided to put 2 castors to be able to stir easily as detailed in Section 3.4

(a) WildTumper 120×60 mm

(b) Castor (Tente™ 2470PJH050P40)

Figure 17: Elements for wheels

### 4.1.4 Obstacle avoidance

The obstacle had two properties that we exploited to the maximum: first they are higher than the bottle, second they are solid. Indeed, as they are higher than the bottles we could have two rows of IR sensors. One lower for the bottles and one higher for the obstacles. Also the obstacle are solid, which means IR sensors are a good choice. For the upper row, we placed three sensors, one in the middle looking forward, one on the left and one on the right, both are a bit tiled, about 20 degrees. Most of the time our robot would avoid the obstacles, however the robot has two blind spots between the sensors (see fig 18) where the robot wouldn't see the obstacles and would hit them. This is quite rare as the robot rarely goes forward for a long time and at some points would turn and see the obstacle.

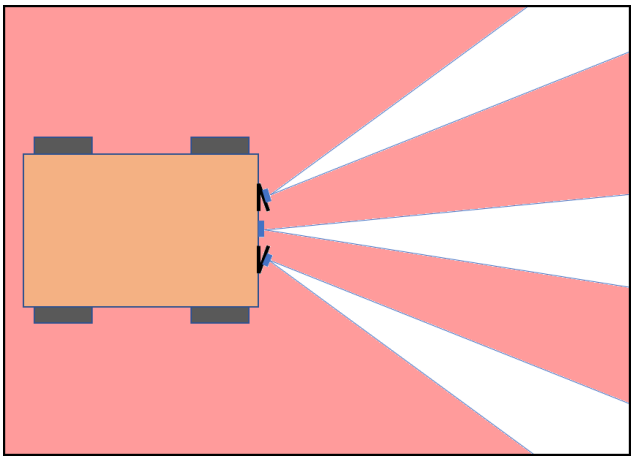
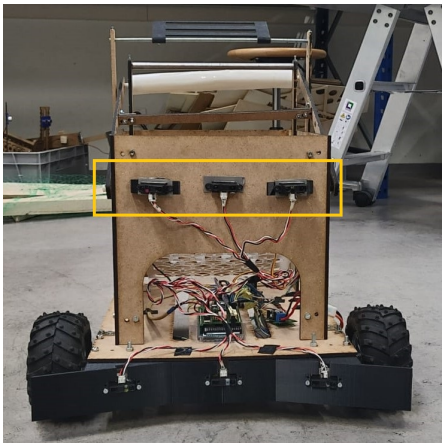


Figure 18: Blind spots of our robot. Red zone: blind spots. White zone: detection cone of IR sensors.



(a) Front view: highlight obstacle detection sensors

(b) Back view: highlight obstacle detection sensors

Figure 19: Highlight obstacle detection sensors

On the other hand we also had two IR sensors in the back. Our robot goes rarely backward, but if he does we want to be sure that it doesn't hit anything. These sensors are also useful when the robot goes back to the recycling area and we want to open the back to release the bottles: we double check that the wall are far enough so that we have enough space.
All our sensors are 80 cm IR proximity sensor. We had the choice between 30 and 80 cm but our final choice was the second one as 30 cm can be too short to react.

#### 4.1.5 Bottle detection

The main idea that we agreed on for the bottle detection was to distinguish the bottles from obstacles via their height differences. Initially, we though that it would be possible to acquire a fairly precise measurement of the distance to the bottles via IR sensors. With it, it would be possible to position the robot in front of the bottle and catch it. However, after testing the IR sensors on bottles, it was clear that the acquired distance values were inconsistent when the bottle type, the angle and the distance of the bottle changed. We then tried the same procedure but with ultrasound sensors to no avail as the results were even worse.

Consequently, our design changed into an array that could work using only thresholds and not precise values. It separates space into three domains as seen on figure 20. The blue domain corresponds to the space

where the sensors return very high noisy values. The green space is under the first threshold and is within the range of the bottle detection algorithm. The red domain corresponds to the space that is so close that it exceeded the focal point of the sensors and is therefore lower than the second threshold.

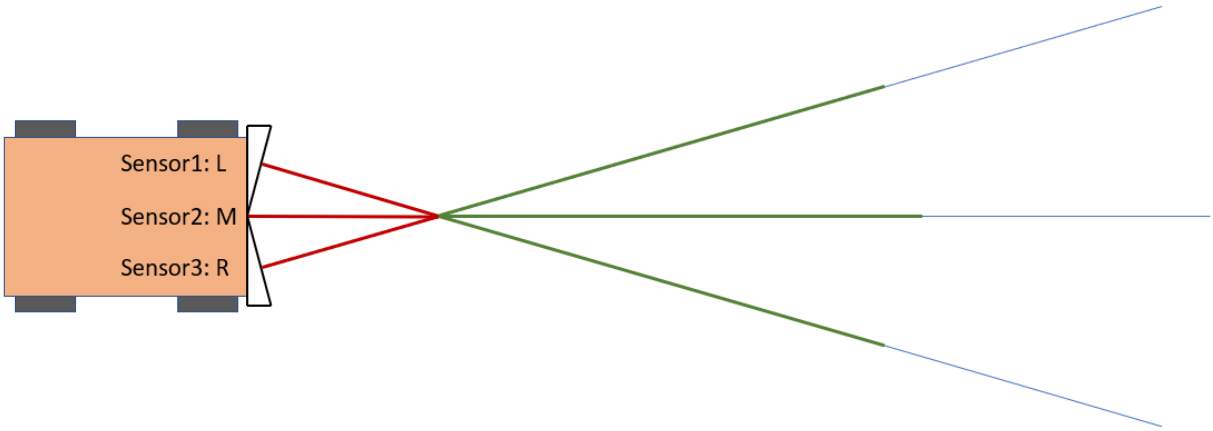



Figure 20: Schematic of the bottom sensors array

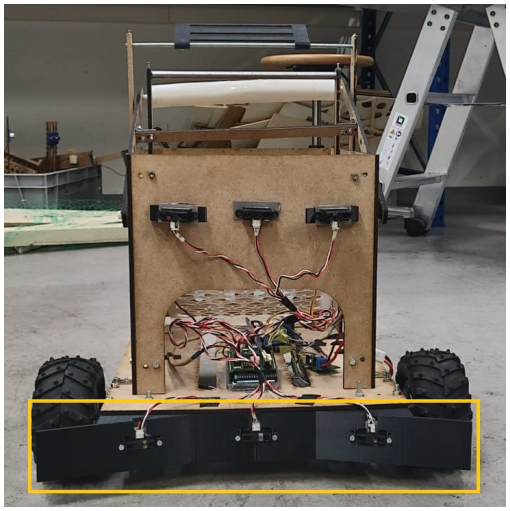


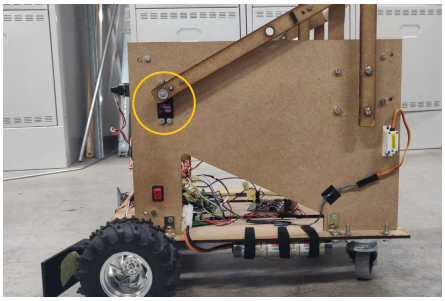
Figure 21: Front view: highlight of bottle detection sensors array

The support for the low sensor array is 3D printed and taped underneath the chassis with VHB™ Tape GPH-110GF. It serves several purposes:

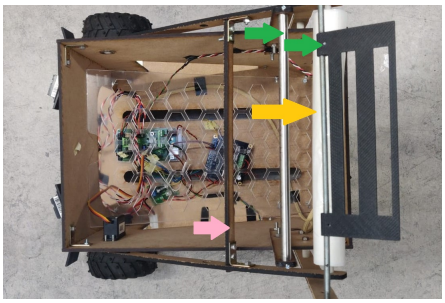
- Hold the IR sensors in place (position & orientation)
- Protect and prevent the wheels from driving over bottles, which could lose the odometry
- Push eventual bottles on the way back to the recycling area (since the bottle detection is disabled while returning home)

### 4.1.6 Bottle grabbing

At this point you are aware that our robot uses an arm to grab the bottle. An easy choice to fit our needs was to use a servo motor strong enough to grab the bottle and to unhooked them so that they fall in the robot. We used a Tower Pro MG995R servo as it was available in the catalog and was perfect for us. However we had to tweak a little the servo library as we noticed that the hardware limits of the servo were not the same as the software limits (i.e the arm could go down more than what the library would allow) . A quick change of value of variable fixed it.
The system is the one detailed in Section 3.1. One can see its final implementation in Fig.22,23. On top of the highest stop we added an extension (black 3D printed piece) in order to prevent the bottles to fall behind the robot during detaching (see Fig.23b).

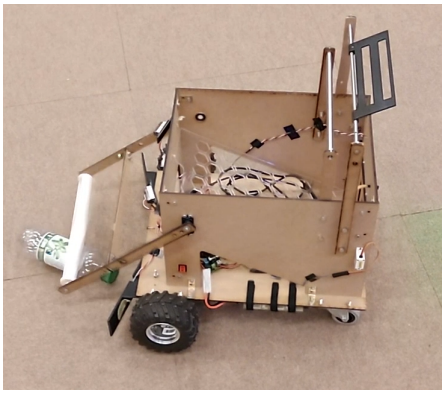


(a) Left view: highlight of arm servo placement

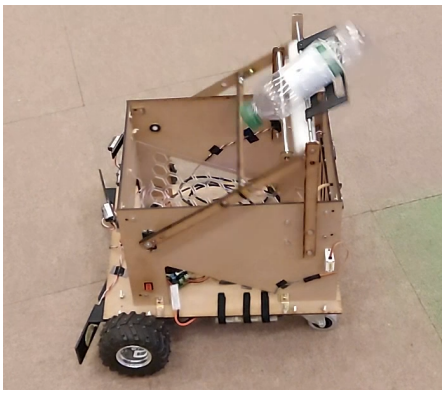


(b) Top view: highlight of arm components
Pink: reinforcement of arm frame. Green: 2 stops.
Yellow: sticky tube

Figure 22: Highlight of bottle grabbing mechanism



(a) Arm down to grab the bottle



(b) Arm up against the stops to unhook the bottle

Figure 23: Bottle grabbing sequence

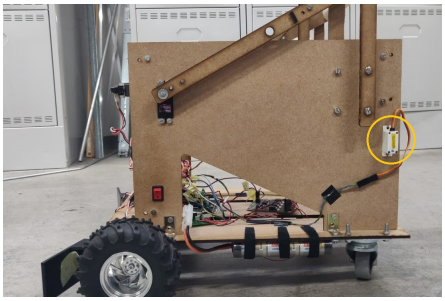
### 4.1.7 Bottle delivery

We had a slope inside the robot and a little door in the back ready to open in order to release the bottles. Thus a small servo to open and close the door was an easy choice. At the beginning we used a TowerPro MG90s and designed the robot's wall according to its dimensions. However, later on, we fried it which lead us to choose another model with similar performances. We went for a micro servo 2.8kg HK15148B which satisfied us quickly. We did not have time to laser cut an entire other wall, so the hole for its placement was adapted with a file. A few days before the competition, during our test the HK15148B fried again because of an inattention error. Luckily, there several identical models in the catalog wich allowed us to quickly replace it, thanks to M. Crespi.

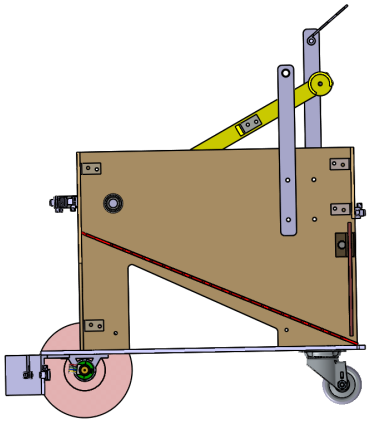
We designed the slope so that there was no chance for the bottle to stay stuck in the robot during the delivery phase:

- It is made of PMMA which a lower coefficient of friction with PET than MDF
- Honeycomb shapes are cut to decrease friction even more
- Its angle with respect to the chassis is large

(a) Back view: highlight of back door. Blue: Door pivots. Yellow: Back door



(b) Left view: highlight of back servo placement



(c) Interior view: highlight of slope placement (in red)

Figure 24: Highlight of bottle delivery mechanism

## 4.2 Software

Our code is written entirely on Arduino and is accessible on GitHub. The base of our code is very basic: go on a random search around the arena. If an obstacle or a wall is blocking the way, avoid it. If a bottle is on the way enter the algorithm to grab it (see 25). This goes on for a given time. After this the robot decides to come back to the recycling area, not looking at new bottles but still avoiding obstacles. Once at the corresponding area, the robot turns around until is has a good angle (looking at the arena and thus looking back at the recycling area) and releases the bottles. It then enters the re-calibration part before going for another run.

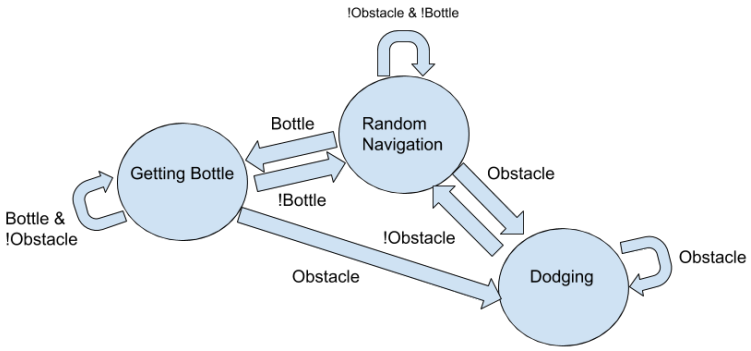



Figure 25: Final State Machine of the robot

### 4.2.1 Motor control

In order to use the motors we had to use both a PID (proportional–integral–derivative )controller and two encoders, one on each motor. The encoder allowed us to read the number of pulse of the motor which gets converted to a distance travelled by the wheel. We then get the speed from it because we know the last time we computed it which was recent enough that we can average it (computed at about every $\Delta t = 40ms$) with $v = \frac{d}{\Delta t}$. This is used for the feedback loop of the PID to calculate the error on the requested speed. The goal of our PID controller is to tweak the PWM (Pulse With Modulation, value given to the motor between 0 and 255) to have the speed read get as close as possible to the wanted speed. The PID has three parameters/variables (Proportional, integral and derivative) that are difficult to find so that they are optimal.

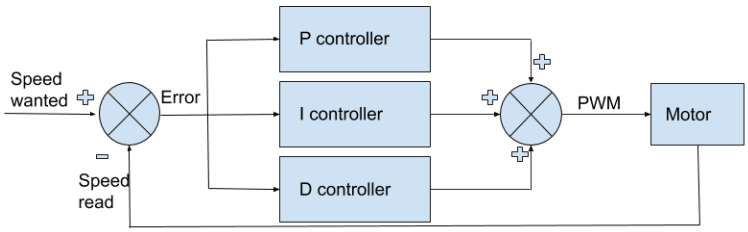



Figure 26: PID

### 4.2.2 Localization

The localization is a core part of our robot. The motors came with encoders on them and inspired us to use the odometry. However it is known that the odometry alone is not enough. As explained earlier we tried to use a compass to adjust it but this did not come with success. In the end we decided to use our robot on a short period of time as the longer the robot runs, the more imprecise the odometry becomes. We've created a recalibration process that is explained in detail later in Section 4.2.6.
As explained in the Section 4.2.1 the encoders on the motors can give us the distance travelled, since the last time we called the function $odometry()$, by each wheel $d_{left}$ and $d_{right}$. The distance travelled by the center of the two wheels becomes $d_{center} = \frac{d_{right}+d_{right}}{2}$. After that we compute the change of the angle of the robot $\phi = \frac{d_{right}-d_{right}}{d_{baseline}}$ where $d_{baseline}$ is the distance between the left and right wheels. Our new angle is then : $\theta' = \theta + \phi$. In the end we can compute the new position of our robot since last time we called the function $odometry()$ with : $x' = x + d_{center} * cos(\theta')$ and $y' = y + d_{center} * sin(\theta')$ [2]. The angle is in radian and goes in the trigonometric direction. An angle of 0 means the robot is looking at the grass area and an angle or $\frac{\pi}{2}$ means you are looking at the rock area. We started our robot at an angle of $\frac{\pi}{4}$ to optimize the navigation.
Most of the noise that get added to the odometry comes from the imprecision of the angle $\theta'$ which as the main reason why we wanted to add a compass in the first place.

### 4.2.3 Navigation

If our robot doesn't see any bottles, it should still be able to go around in order to see a bottle. Even if this seems trivial, we still need a good approach for it. Thus we went for a random navigation. For any position $(x, y)$ our robot would predict three future position at an angle of $\frac{\pi}{4}$, 0 and $\frac{-\pi}{4}$ and a given radius. If one of the position was not valid, the robot would not consider going to this position for example the first prediction at the figure 27. For the others, it computes again three predictions in order to have weights on the probability on which prediction to chose. For example on the figure 27 the second prediction has then only two predictions where the robot will be able to go to whereas the third prediction has three, thus the robot has a probability of $\frac{2}{(2+3)} = \frac{2}{5}$ to go forward and a probability of $\frac{3}{5}$ to go on the right (and of course 0 to go on the left because this is outside of the arena). On the other hand, if out of every predictions of every 3 predictions none is valid, the robot would go backward. We make sure that after going backward the robot doesn't go forward again not get to the same position.

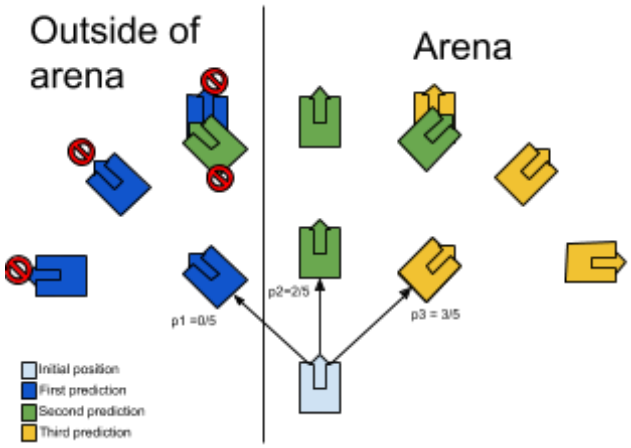



Figure 27: Algorithm of navigation

However as our PID controller did not had the perfect parameter the robot would not quite reach the wanting position but a bit less which was fine for us. We created this algorithm with the idea that this would encourage the robot to go away from the walls and from the zones we don't want to go (rocks, grass, the upper part or the area and of course outside of the arena). On average the robot would move forward and we would give him an initial angle of $\frac{\pi}{4}$ so that it would explore the center of the arena. It is to remember that if the robot sees any obstacle or bottle, it goes out of the navigation part and enter the corresponding one (see fig 25).
An example of this navigation algorithm can be seen on 28. The robots starts at the position $(0, 0)$ and goes around the arena, after some iterations it tries to come back to the recycling area (this can be seen as the straight line). On this simulation there is no obstacle or bottle, this is only to show the navigation algorithm.

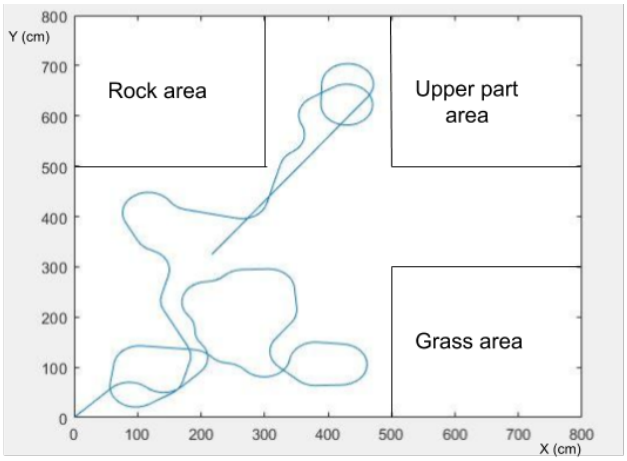



Figure 28: Example of navigation

This navigation algorithm was quite useful as we couldn't see the rocks or the grass as obstacle with our sensors. With this, the robot would naturally avoid going into these zones and with higher probability get further away from them.

#### 4.2.4 Bottle detection

To move the robot towards the bottle's location, a logic table was designed and then implemented in a series of if statements. The following table show the movement decisions in the sensors' green domain (cf figure 20).

Table 8: Logic table for the robot movement in the green domain

| M=0 | | | | M=1 | | | |
|---|---|---|---|---|---|---|---|
| | | R | | | | R | |
| | | 1 | 0 | | | 1 | 0 |
| L | 1 | impossible scenario | turn right | L | 1 | bottle is in place | turn right slowly |
| | 0 | turn left | nothing is detected | | 0 | turn left slowly | move forward |

These movements were designed in a way that the robot converges on the bottles. If the signals given by the sensors become inferior to the second threshold, namely that the bottle is located in the red domain, the movements that were going right go left and vice versa. This is because the bottle has exceeded the focal point of the sensors. Once all the sensors detect a bottle a sequence starts: the robot moves forward a little bit to make sure that the bottle is right in front of him, he then proceeds to grab the bottle with the arm. Once the robot has grabbed the bottle, it returns to its usual navigation routine. If the robot did not manage to catch the bottle, it will detect it again as a new bottle and repeat the alignment procedure (though it should already be very close to the grabbing point). This algorithm only changes the direction the robot is going and to catch a bottle, the robot will have to go through this function many times (until it reaches the bottle).

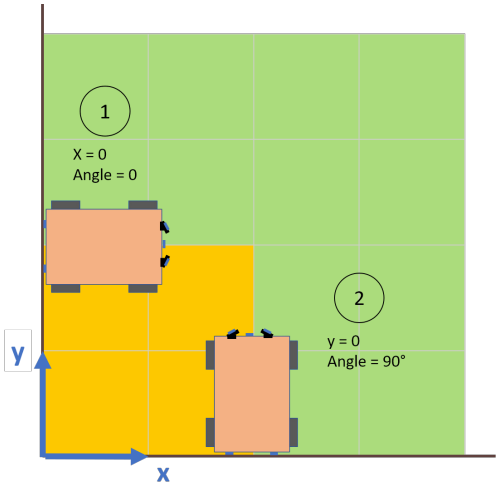
Within the bottle detection algorithm, there is also a Boolean that shows it is in a bottle detecting state (mainly used for localisation), some odometry refreshes and some PID controller refreshes to ensure that the robot is moving correctly and knows where it is. Another important feature that combines the bottle detection and the navigation algorithms is the grey areas. These grey areas surround zone 2 and 3 and are one meter thick. In these areas, the robot will not enter the bottle detection function and will remain in normal navigation. This was implemented in response to the fact that the robot cannot differentiate between rocks, grass and bottles (since their heights are similar). This feature only activates when the robot is facing the zones in question, so there is an additional condition on the angle of the robot.

### 4.2.5 Obstacle detection

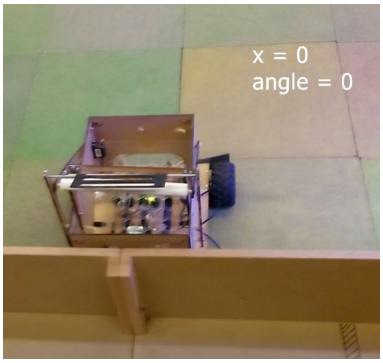
Using the three IR sensors we had to come with an efficient algorithm to dodge the obstacle. We came up with the following: If the two sides sensors, or the three sensors, would see something, go backward. If only one of the sides sensors senses an obstacle, dodge in the opposite direction (ex: if the left senses, dodge by the right). If only the front sensors detects something, go backward.
This is pretty simple but it achieved the results we wanted.

### 4.2.6 Recalibration

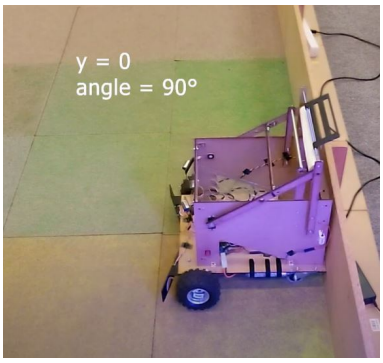
As we only had the odometry for the localisation of our robot we had to come up with something on top of it, as the odometry get more and more noisy as time goes on (Fig.9). We needed an absolute localisation where we were sure the robot was at the correct position. To do so, we used the walls of the corner of the recycling area. Indeed, once the robot got the bottle out, it would turn it back to one of the wall, move backward until it touches the wall. Once this was done, the robot knows his absolute angle and one of his coordinate. What's left to do it to do the same on the other wall and the robot will know both his coordinates and his angle and it will be ready for another run.




(a) Schema recalibration




(b) First recalibration




(c) Second recalibration

Figure 29: The recalibration process

## 4.3 Summary

This section aims to summarize our choices for one to have a quick overview of GaliPET's features.

Table 9: Summarising table of the robot features and solutions

| Feature | Solution | Implementation |
|---|---|---|
| Power | Battery | NiMH battery (7,2V and 3000mAh) |
| Locomotion | Differential robot | 2 Pololu 75:1 DC motor with encoder (HP)<br>2 castor wheels<br>1 H-bridge motor driver<br>Software based PID controller |
| Localisation | Odometry | Odometry software using the encoders of the DC motors |
| Navigation | Random search | Random search using the prediction of its future possible positions as bias for route choice. This takes into account the arena layout. |
| Obstacle avoidance | IR sensor array | Three elevated sensors on the front of the robot and two on the back combined with a dodging maneuver and the route selection of the navigation |
| Bottle detection | IR sensor array | Three converging sensors close to ground level in the front of the robot with a movement choice function to converge on bottles |
| Bottle grabbing | Sticky arm | Servo connected to an arm equipped with a cylinder covered in double sided tape. The cylinder goes through two static bars to remove the bottles |
| Bottle delivery | Reservoir with opening door | The navigation is used to head back to the recycling point. The robot opens the back door and the bottle slide on the reservoir slope |
| Recalibration | Adjustment of odometry | The robot pushes himself on the walls to adjust his position and angle after bottle delivery. |

# 5 Management

## 5.1 Time management

As we were a team of three people, it was impossible for everybody to be there at the same time all the time. Consequently, we came as our time tables allowed it. To compensate this somewhat chaotic timetable, we set some weekly goals up so that our workload could be predicted and monitored. The following figure shows our initial Gantt chart :

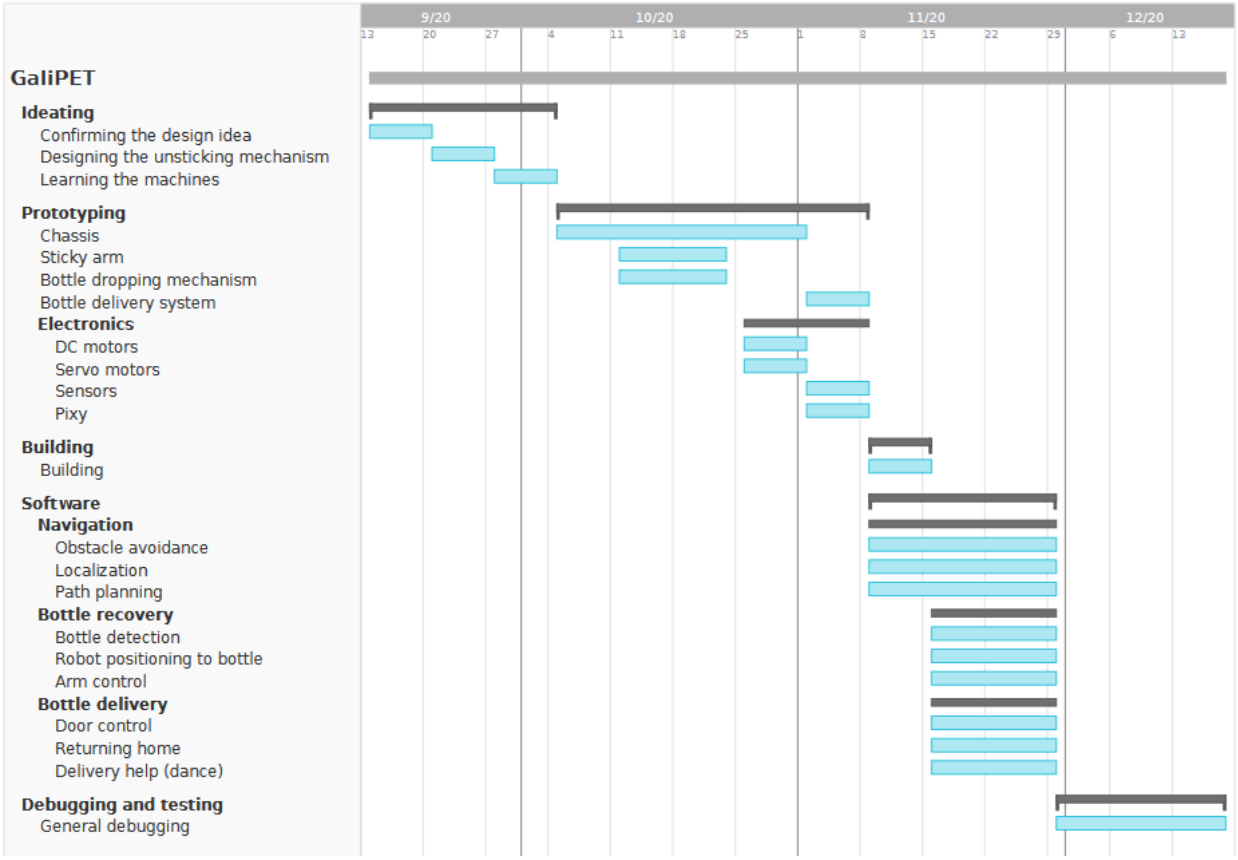



Figure 30: Initial Gantt chart of our project

This schedule was slightly too optimistic as the software part was more time consuming then originally planned. We therefore had to cut on our debugging time and integrated it in the software development so it was done in parallel. Even though we did encounter a number of unexpected problems, our time management allowed us to finish on time for the competition without any extreme measures.

## 5.2 Team management

In terms of team management, it was well coordinated. Our human relationships were good as we knew each other beforehand. The communication was fluid. We quickly set up a WhatsApp group as well as a shared file on GitHub. For work distribution, we did not have any rigid plan. The tasks were given along the way according to their work requirements and the relative affinity of the receiver to the task in question. We believe it lead to well balanced and enjoyable work conditions.

## 5.3 Process management

During the process of this project we have worked with numerous people. This first one being our coach Özdemir Can Kara. We organized weekly meetings with him at the start of the project so that we could start off in the correct direction. He has given valuable advice especially on what to prioritize our work on. The workshops available to us were a great help. Since we had no prior experience in robot building, we had to receive formations for the machines that we used in the project, most notably the laser cutter in the SKIL. Finally, experts have helped us when we presented them with some of our problems. André Hodder was contacted when we struggled to make the DC motors work. Alessandro Crespi has helped us with our numerous electronics questions as well as the distribution of all the parts that we ordered. We are thankful for all of their fast and helpful responses.

## 5.4 Budget management

Since the goal of out project was to keep our design as simple as we could think of, the budget that was required to create was not the most extravagant. Indeed, the initial budget that we received to create this robot was : 750 CHF of real budget and 2000 CHF of virtual budget.
The final budget that we planned to take as of Milestone 3 was around 850 CHF. The final budget that we actually used was 933.52 CHF of which 864.87 CHF were from the virtual budget and 68.85 were from the real budget. This is a bit higher than expected but we had some unfortunate part failures. That being said, the final budget is still very well under the allocated budget.

Tables 10, 11, 12 & 13 summarize the cost of the robot with **only** the part mounted on the final version of GaliPET, the one that ran the competition. This budget is far from the one mentioned above because of many parts we did not use in the end such as spare compass or the pixy camera which are quite expensive material. In addition, there were many tries and errors during the 3D printing processes that lead to an increase of the 3D printing budget. In the end, the total cost of only the part mounted on the robot for the competition, without taking into account the prototyping process with its tries and errors, is 502.98 CHF.

Table 10: Virtual budget

| Description | Price/piece | Quantity | Total |
|---|---|---|---|
| Arduino Mega2560 R3 | 44,5 | 1 | 44,5 |
| Tower Pro MG995R digi hi torque | 10 | 1 | 10 |
| WildTumper 120x60 mm wheel, 4 mm shaft | 7,5 | 2 | 15 |
| DFRobot DRI0018 2x15A motor driver | 45 | 1 | 45 |
| 75:1 DC motor with encoder (HP) | 36,95 | 2 | 73,9 |
| 80 cm IR proximity sensor | 10,95 | 8 | 87,6 |
| micro servo 2,8kg HK15148B | 5 | 1 | 5 |
| NiMH rechargeable battery pack (7,2 V, 3000 mAh) | 19,95 | 1 | 19,95 |
| 3D printing | 111,88 | 1 | 111,88 |
| Total | | | 412,83 CHF |

Table 11: Virtual budget: 3D printing detail

| Name | Price |
|---|---|
| Pack_CYLINDRE_COLLE_groupe_3 | 20,07 |
| Pack_PALIER_MOTEUR_AV | 4,58 |
| Pack_PALIER_MOTEUR_AV | 9,01 |
| Pack_PALIER_MOTEUR_ROULEMENT_GROUP_3 | 17,05 |
| Pack_PIVOT_MOTEUR | 5,99 |
| Pack_PIVOT_PORTE_GROUP3 | 6,38 |
| SUPPORT_SENSORS_AVANT | 20,07 |
| SUPPORT_SENSORS_LEFT | 19,98 |
| SUR_BUTTEE_1 | 8,75 |
| Total | 111,88 CHF |

Table 12: Real budget

| Description | Price/piece | Quantity | Total |
|---|---|---|---|
| Impression PETG | 2,1 | 1 | 2,1 |
| 2 plaques MDF 4mm + 1 plaque plexi 3mm | 11,4 | 1 | 11,4 |
| Commande Distrelec du 17,11,2020 | 50,15 | 1 | 50,15 |
| Fried servo | 5 | 2 | 10 |
| Castors (OBI) | 8,25 | 2 | 16,5 |
| Total | | | 90,15 CHF |

Table 13: Total price

| Budget | |
|---|---|
| Virtual | 412,83 CHF |
| Real | 90,15 CHF |
| Total | 502,98 CHF |

# 6 Competition results

The results of the competition were that our robot managed to grab 3 bottles using 3 resets to the starting area. The bottles were all from zone 1, our only target area. The navigation and the bottle detection functions worked well. However, the grabbing of the bottle and specifically the unsticking of the bottles was pretty unlucky during the competition as 3 bottles that were grabbed fell outside the reservoir when they were unsticked. Additionally, we encountered an event that got our robot stuck in a way that we had never experienced during all our testing : a bottle got stuck under our robot leading to a reset. To conclude, we are still overall proud of our robot because we accomplished the task we set out to do, namely, to build from scratch an autonomous robot that can grab and deliver bottles.

# 7 Learnings of this project

## 7.1 Improvements for GaliPET 2.0

There are a few improvement that we could have done with some more time. Adding a Raspberry Pie would have given us the option to have a camera and a better bottle and obstacle detection. We tried a for a long time to make to compass worked but eventually we could have tried to use an IMU as different option to recalibrate the odometry.
We could have also added another servo for the arm so that the arm would more powerful and more precise. We've added a 3D piece on the arm to reduce the chances for the bottle to get out of the robot when they are un-taped from the arm. However some bottle would still get out. Still could be fixed with some higher wall on the sides of the robot for example.

## 7.2 Hints for future teams

From our experience we can pass on a few tips for future teams:

- Our coach and experts are here to help us, so do not hesitate to question them
- Prototype your ideas as soon as possible. Their qualities and flaws will be quickly highlighted in doing so
- Schedule your project to be ready to test your robot as soon as the final arena is mounted. Nothing will teach you about your robot more than those tests
- This is a unique project ! Have fun !

# 8 Conclusion

To conclude, our Robot is still not perfect and it could benefit from some improvements. However, we are still proud of the results of this project. We made a functional robot. Moreover, we believe that the most important aspect of this project is its learning outcomes. The quantity of skills we have developed is incredible. This includes some transversal skills in project management but mainly numerous practical skills such as laser cutting, 3D printing, soldering, mechanical design, prototyping, programming and so on.

# References

[1] Fox, D., Burgard, W., Dellaert, F., & Thrun, S. (1999). *Monte carlo localization: Efficient position estimation for mobile robots*. AAAI/IAAI, 1999(343-349), 2-2.

[2] Olson E. (2004). *A Primeron Odometry and Motor Control*, MIT - OpenCourseWare. Link to article Last consulted: january 2021

# A Presentation video

We made a presentation video showing all the features of our robot:

- Localization and navigation
- Obstacle avoidance
- Moving bottles & delivery to recycling area
- Recalibration of odometry

# B Electronics drawing

The main core of our robot was the Arduino where we've put on top an Arduone. Most of the components were directly connected to the Arduino such as the servos and the sensors. The servos were connected to the PWM (Pulse With Modulation) pins and the sensors to the Analogs pins. As explained before the Arduino was powered using an NiMH battery. The Motor Driver was also powered using an NiMH battery and would received one PWM value (between 0 and 255) for each motor from the Arduino. The encoders are also connected to Arduino as they are giving feedback on the distance travelled by each wheel.

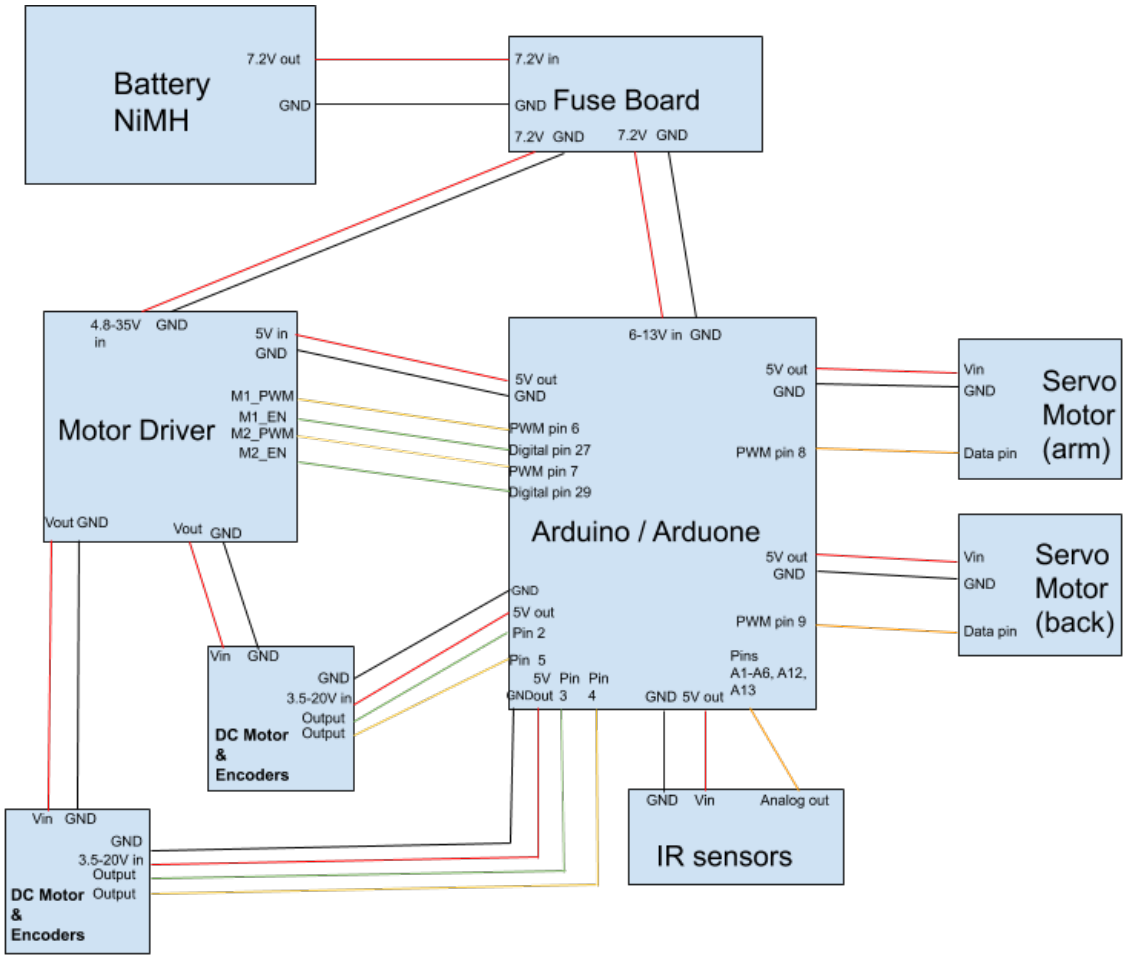
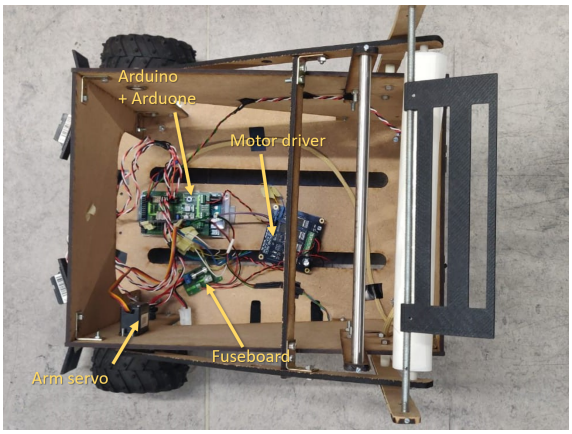
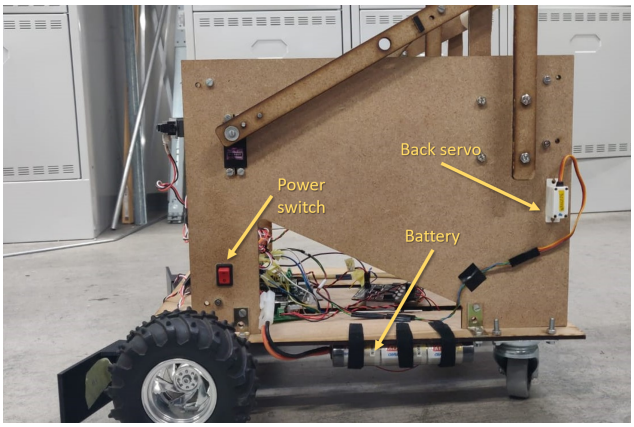


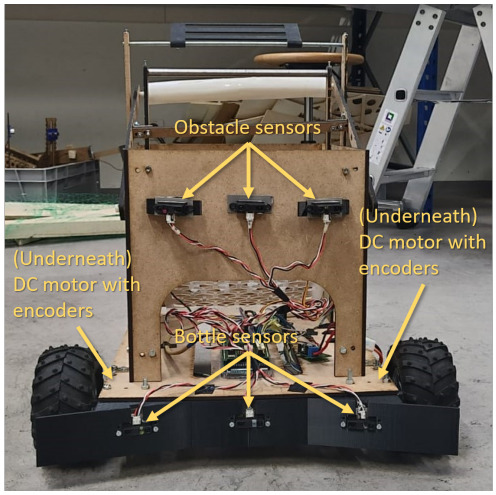

Figure 31: Electronic Drawing




(a) Top view without slope: overview of inside electronics




(b) Left view: overview electronics

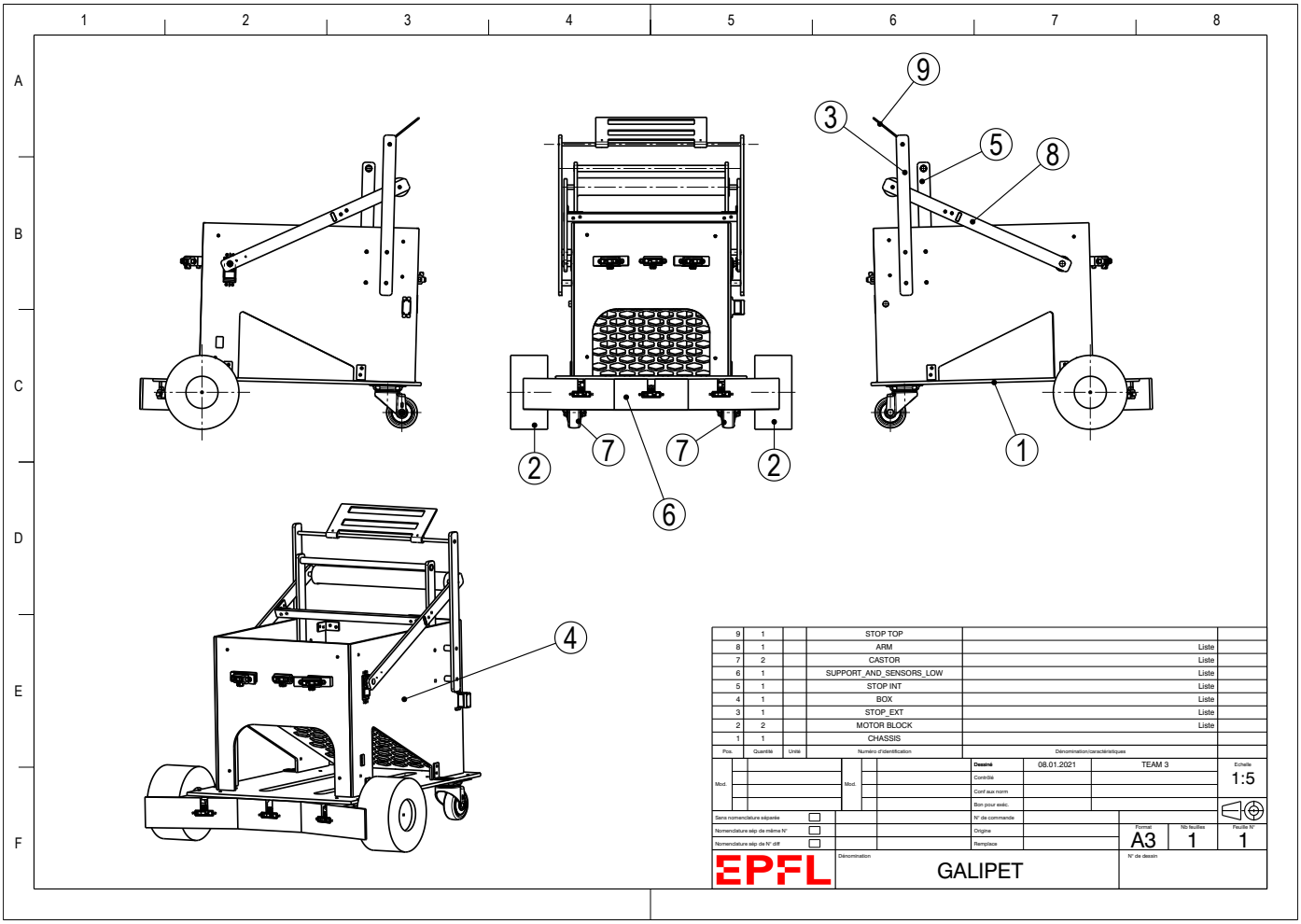



(c) Front view: overview electronics

Figure 32: Overview of electronics on GaliPET

# C Mechanical drawings

If not specified otherwise, all the the parts' dimension tolerances correspond to the norm ISO 2768-mK.




| Pos. | Quantité | Unité | Numéro d'identification | Dénomination/caractéristiques |
|---|---|---|---|---|
| 9 | 1 | | STOP TOP | |
| 8 | 1 | | ARM | Liste |
| 7 | 2 | | CASTOR | Liste |
| 6 | 1 | | SUPPORT_AND_SENSORS_LOW | Liste |
| 5 | 1 | | STOP INT | Liste |
| 4 | 1 | | BOX | Liste |
| 3 | 1 | | STOP_EXT | Liste |
| 2 | 2 | | MOTOR BLOCK | Liste |
| 1 | 1 | | CHASSIS | |

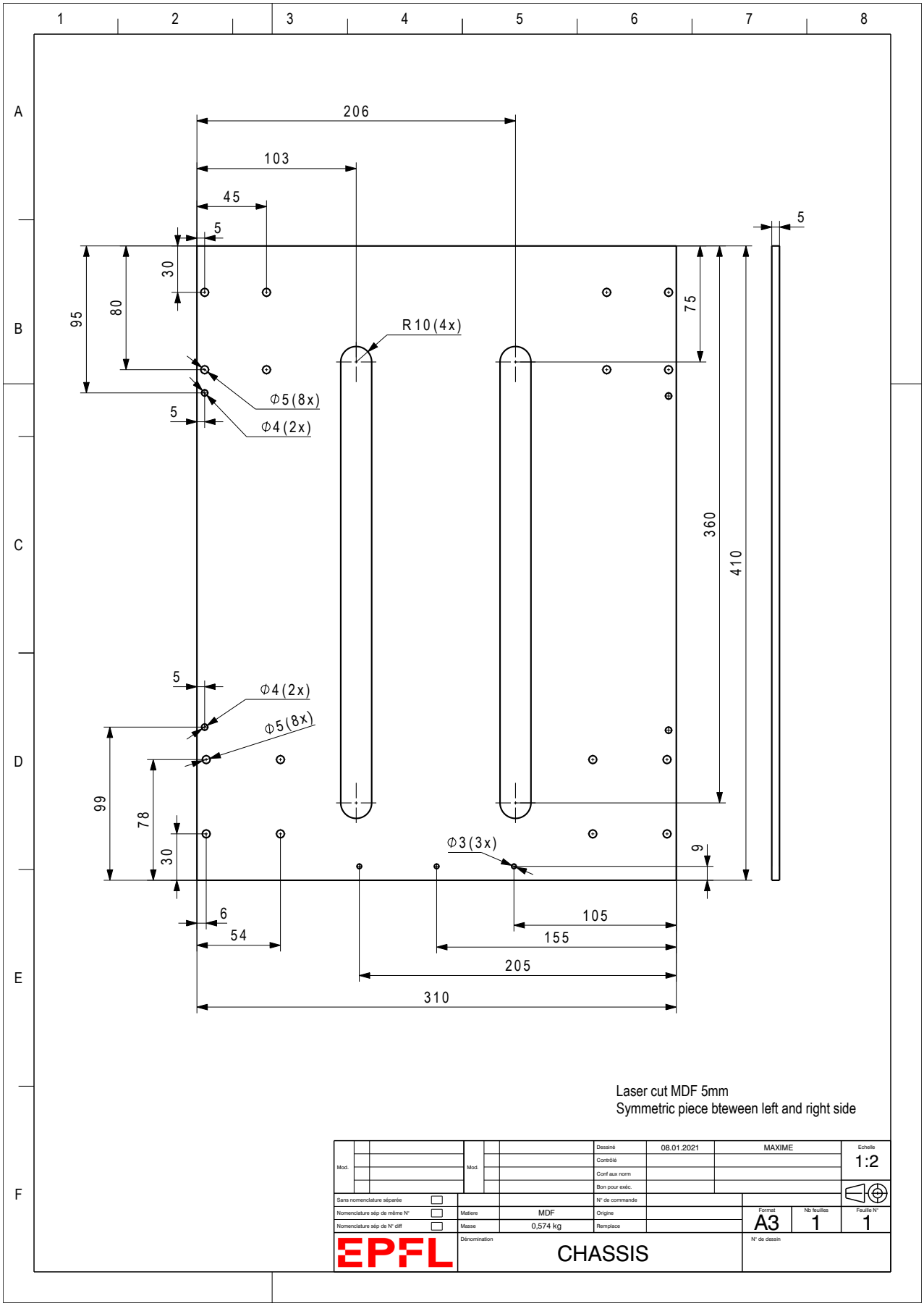
Mod. | Mod. | Dessiné 08.01.2021 | TEAM 3 | Echelle 1:5

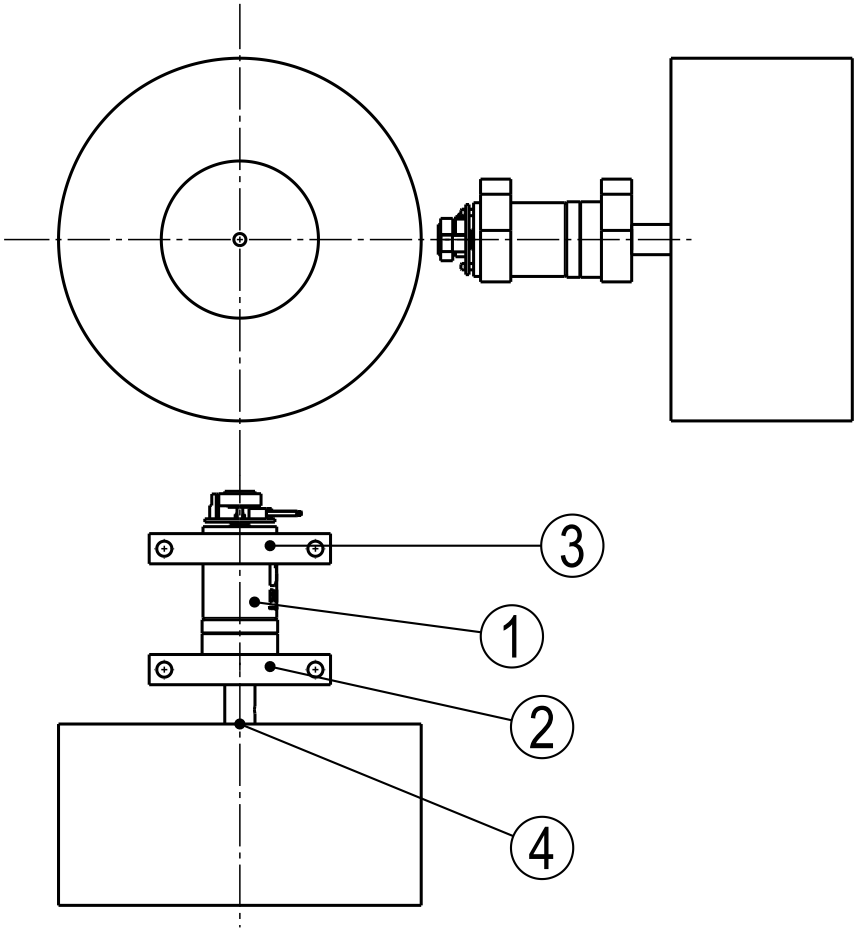
Contrôlé | Conf aux norm | Bon pour exéc.

Sans nomenclature séparée | Nomenclature sép de même N° | Nomenclature sép de N° diff | N° de commande | Origine | Remplace

Format A3 | Nb feuilles 1 | Feuille N° 1

EPFL | Dénomination: GALIPET | N° de dessin


206
103
45
5
30
95
80
R10(4x)
Φ5(8x)
Φ4(2x)
5
75
5
360
410
5
Φ4(2x)
Φ5(8x)
99
78
30
Φ3(3x)
9
6
54
105
155
205
310
Laser cut MDF 5mm
Symmetric piece bteween left and right side
Dessiné 08.01.2021 MAXIME
Echelle 1:2
Contrôlé
Conf aux norm
Bon pour exéc.
N° de commande
Matiere MDF
Masse 0,574 kg
Origine
Remplace
Format A3
Nb feuilles 1
Feuille N° 1
Dénomination CHASSIS
N° de dessin




| Pos. | Quantité | Unité | Numéro d'identification | Dénomination/caractéristiques | |
|---|---|---|---|---|---|
| 4 | 1 | | WildTumper 120x60 mm wheel, 4 mm shaft | WildTumper 120x60 mm wheel, 4 mm shaft | Liste |
| 3 | 1 | | PALIER_MOTEUR_AR | | |
| 2 | 1 | | PALIER_MOTEUR_AV | | |
| 1 | 1 | | Pololu 2275 | Pololu 2275 | |

| | | | |
|---|---|---|---|
| Mod. / Mod. | Dessiné | 08.01.2021 | MAXIME |
| | Contrôlé | | |
| | Conf aux norm | | |
| | Bon pour exéc. | | |

Echelle 1:2

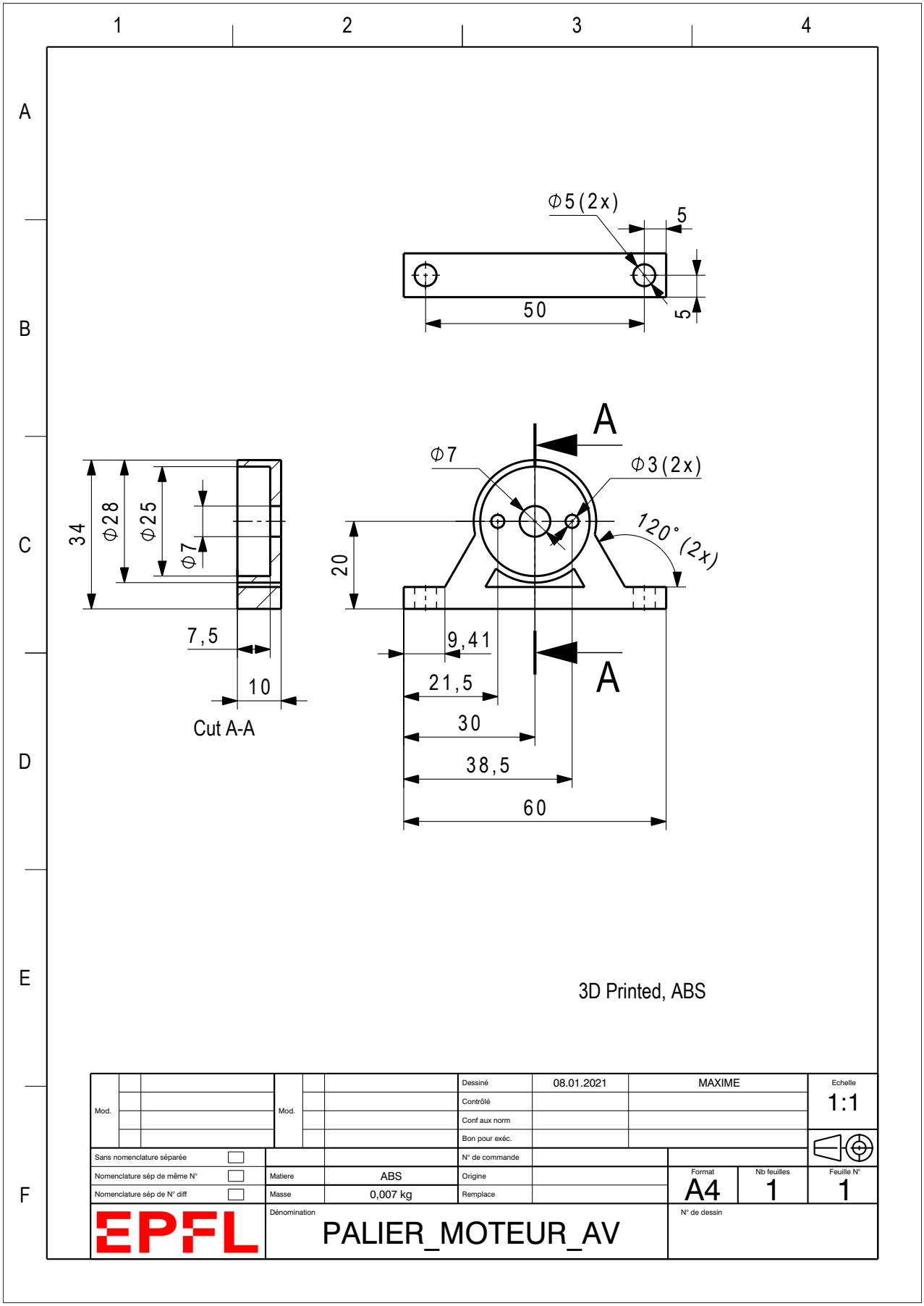
| | | | |
|---|---|---|---|
| Sans nomenclature séparée ☐ | N° de commande | | |
| Nomenclature sép de même N° ☐ | Origine | | |
| Nomenclature sép de N° diff ☐ | Remplace | | |

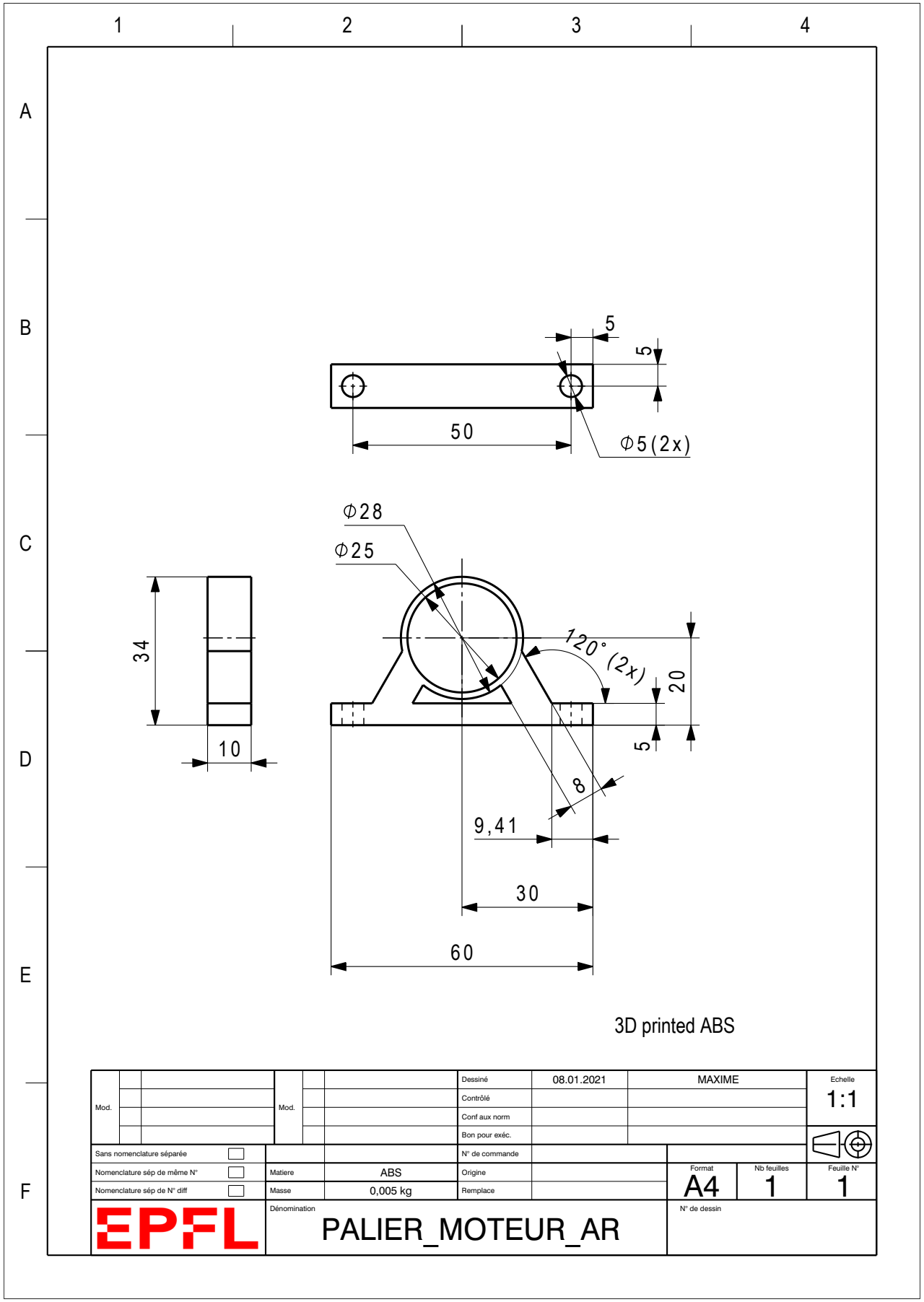
| Format | Nb feuilles | Feuille N° |
|---|---|---|
| A4 | 1 | 1 |

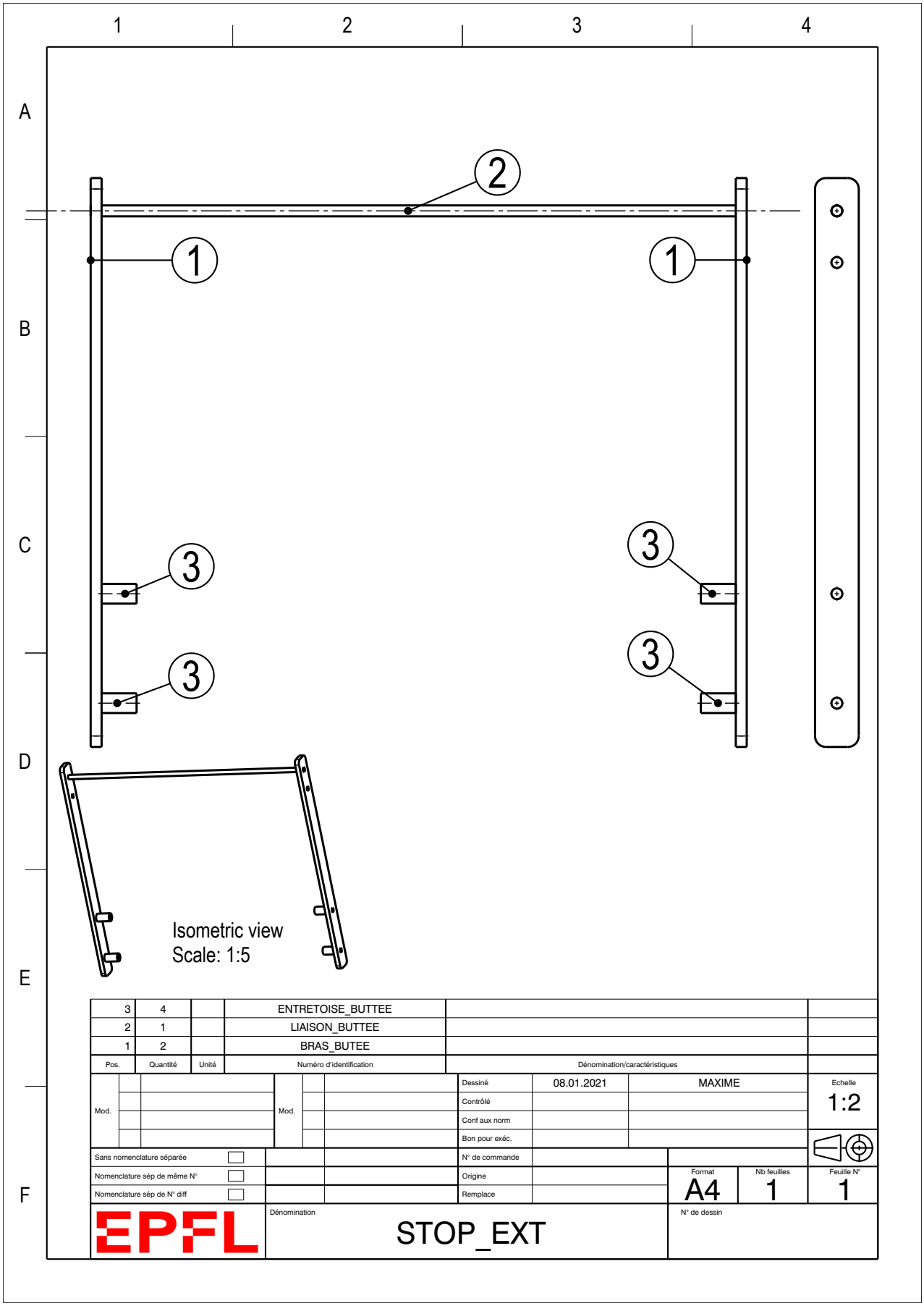
EPFL

Dénomination: MOTOR BLOCK

N° de dessin:


1
2
3
4
A
B
C
D
E
F
Φ5 (2x)
5
5
50
A
Φ7
Φ3 (2x)
120° (2x)
20
9,41
21,5
30
38,5
60
A
34
Φ28
Φ25
Φ7
7,5
10
Cut A-A
3D Printed, ABS
Mod.
Mod.
Dessiné
08.01.2021
MAXIME
Contrôlé
Conf aux norm
Bon pour exéc.
Echelle
1:1
Sans nomenclature séparée
Nomenclature sép de même N°
Nomenclature sép de N° diff
N° de commande
Matiere
ABS
Origine
Masse
0,007 kg
Remplace
Format
A4
Nb feuilles
1
Feuille N°
1
EPFL
Dénomination
PALIER_MOTEUR_AV
N° de dessin


1
2
3
4
A
B
C
D
E
F
5
5
50
Φ5 (2x)
Φ28
Φ25
34
10
120° (2x)
20
5
8
9,41
30
60
3D printed ABS
Mod.
Mod.
Dessiné
08.01.2021
MAXIME
Contrôlé
Conf aux norm
Bon pour exéc.
Echelle
1:1
Sans nomenclature séparée
Nomenclature sép de même N°
Nomenclature sép de N° diff
N° de commande
Matiere
ABS
Origine
Masse
0,005 kg
Remplace
Format
A4
Nb feuilles
1
Feuille N°
1
EPFL
Dénomination
PALIER_MOTEUR_AR
N° de dessin




| Pos. | Quantité | Unité | Numéro d'identification | Dénomination/caractéristiques |
|---|---|---|---|---|
| 3 | 4 | | ENTRETOISE_BUTTEE | |
| 2 | 1 | | LIAISON_BUTTEE | |
| 1 | 2 | | BRAS_BUTEE | |

| | | | |
|---|---|---|---|
| Dessiné | 08.01.2021 | MAXIME | Echelle 1:2 |
| Contrôlé | | | |
| Conf aux norm | | | |
| Bon pour exéc. | | | |

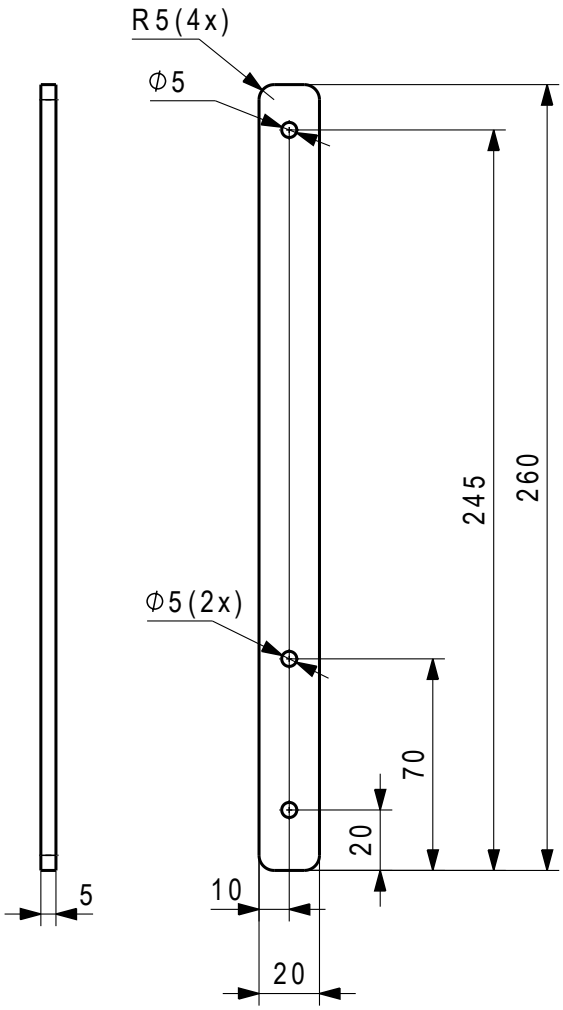
| | | | |
|---|---|---|---|
| Sans nomenclature séparée | | N° de commande | |
| Nomenclature sép de même N° | | Origine | |
| Nomenclature sép de N° diff | | Remplace | |

| Format | Nb feuilles | Feuille N° |
|---|---|---|
| A4 | 1 | 1 |

EPFL

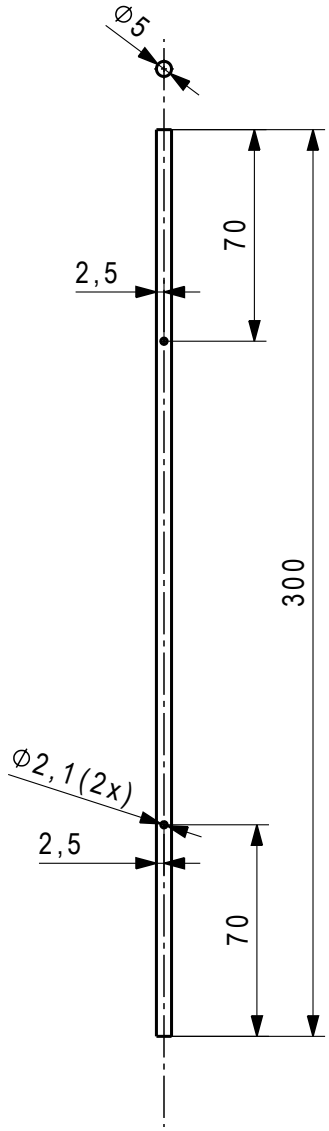
Dénomination: STOP_EXT

N° de dessin:

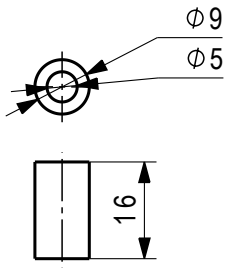



Laser cut MDF 5mm

| | | | | | |
|---|---|---|---|---|---|
| Mod. / Mod. | | Dessiné | 09.01.2021 | MAXIME | Echelle 1:2 |
| | | Contrôlé | | | |
| | | Conf aux norm | | | |
| | | Bon pour exéc. | | | |
| Sans nomenclature séparée ☐ | | N° de commande | | | |
| Nomenclature sép de même N° ☐ | Matiere: MDF | Origine | | Format A4 | Nb feuilles 1 / Feuille N° 1 |
| Nomenclature sép de N° diff ☐ | Masse: 0,019 kg | Remplace | | | |
| EPFL | Dénomination: BRAS_BUTEE | | | N° de dessin | |




| | | | | | |
|---|---|---|---|---|---|
| Mod. | Mod. | Dessiné | 08.01.2021 | MAXIME | Echelle 1:2 |
| | | Contrôlé | | | |
| | | Conf aux norm | | | |
| | | Bon pour exéc. | | | |
| Sans nomenclature séparée ☐ | | N° de commande | | | |
| Nomenclature sép de même N° ☐ | Matiere: 100 Mn Cr W 4 | Origine | | Format A4 | Nb feuilles 1 / Feuille N° 1 |
| Nomenclature sép de N° diff ☐ | Masse: 0,046 kg | Remplace | | | |
| EPFL | Dénomination: LIAISON_BUTTEE | | | N° de dessin | |




3D printed ABS

| | | | | | | |
|---|---|---|---|---|---|---|
| Mod. | | Mod. | | Dessiné | 08.01.2021 | MAXIME |
| | | | | Contrôlé | | |
| | | | | Conf aux norm | | |
| | | | | Bon pour exéc. | | |

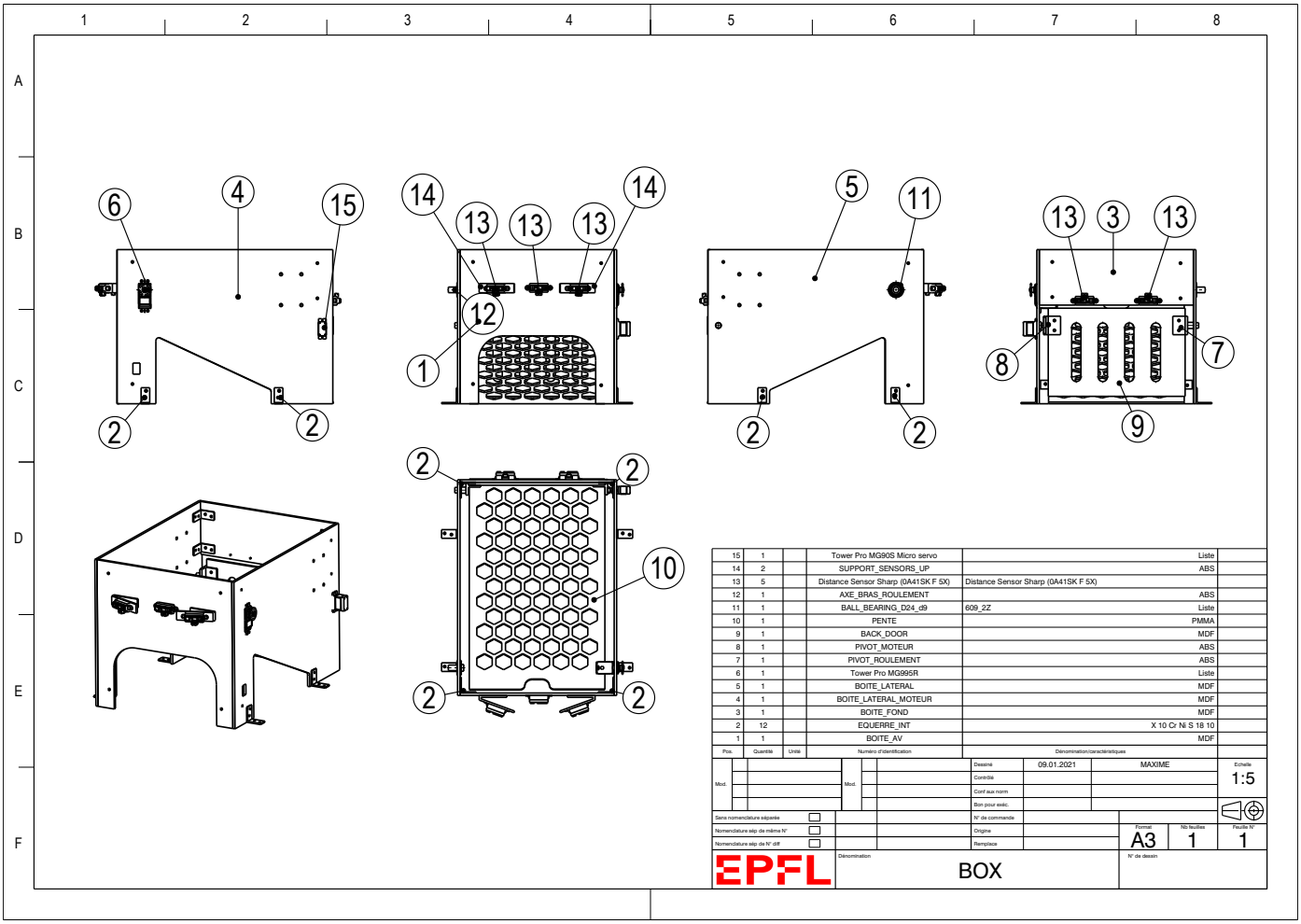
Echelle 1:1

| | | | | | | |
|---|---|---|---|---|---|---|
| Sans nomenclature séparée ☐ | | | N° de commande | | | |
| Nomenclature sép de même N° ☐ | Matiere | ABS | Origine | | Format A4 | Nb feuilles 1 |
| Nomenclature sép de N° diff ☐ | Masse | 0,001 kg | Remplace | | | Feuille N° 1 |

EPFL

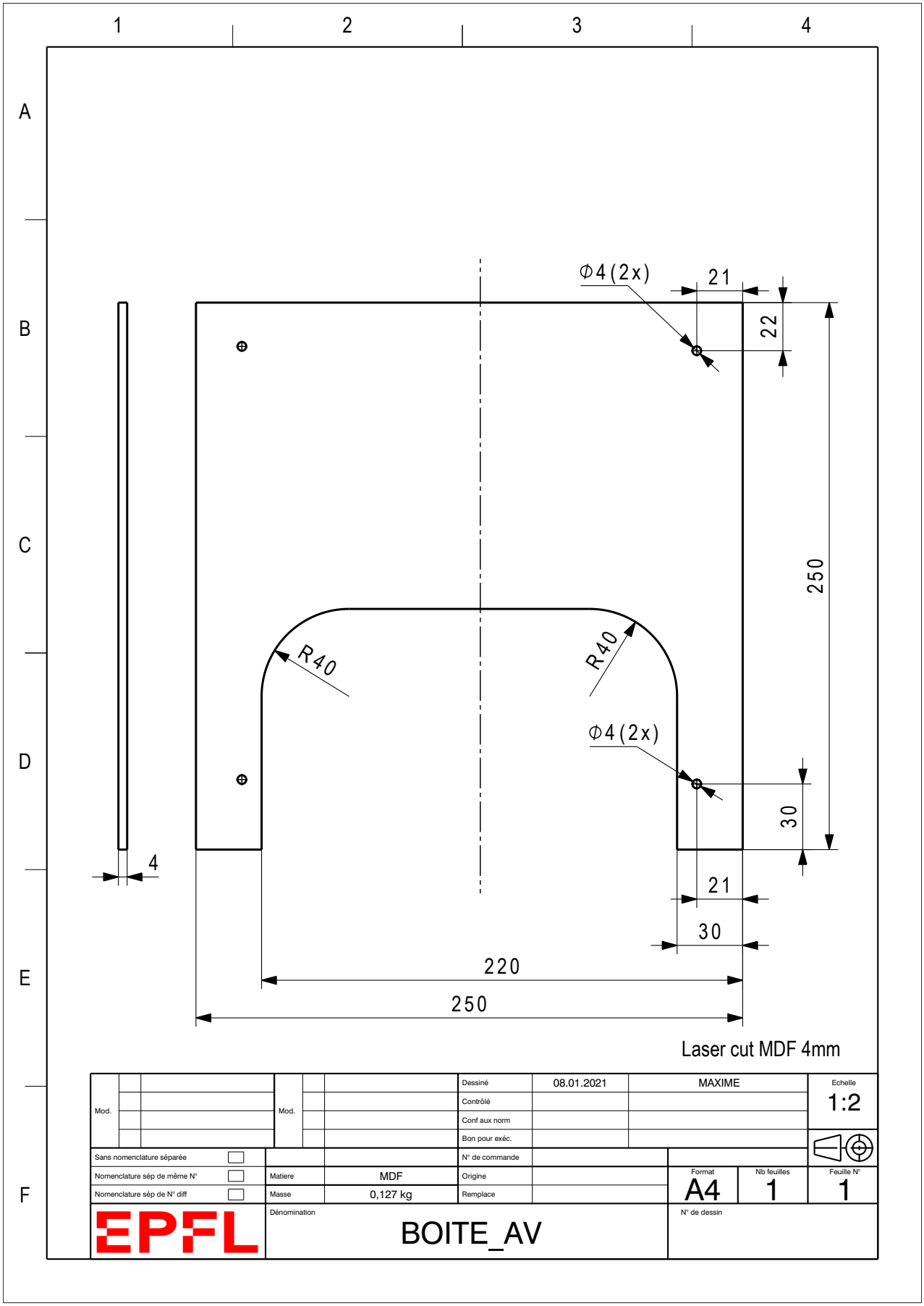
Dénomination: ENTRETOISE_BUTTEE

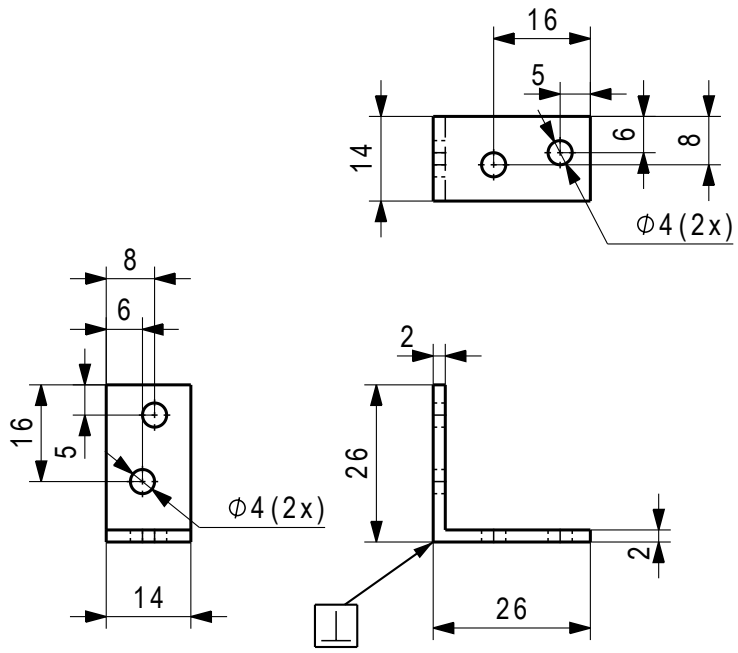
N° de dessin




| Pos. | Quantité | Unité | Numéro d'identification | Dénomination/caractéristiques |
|---|---|---|---|---|
| 15 | 1 | | Tower Pro MG90S Micro servo | Liste |
| 14 | 2 | | SUPPORT_SENSORS_UP | ABS |
| 13 | 5 | | Distance Sensor Sharp (0A41SK F 5X) | Distance Sensor Sharp (0A41SK F 5X) |
| 12 | 1 | | AXE_BRAS_ROULEMENT | ABS |
| 11 | 1 | | BALL_BEARING_D24_d9 | 609_2Z Liste |
| 10 | 1 | | PENTE | PMMA |
| 9 | 1 | | BACK_DOOR | MDF |
| 8 | 1 | | PIVOT_MOTEUR | ABS |
| 7 | 1 | | PIVOT_ROULEMENT | ABS |
| 6 | 1 | | Tower Pro MG995R | Liste |
| 5 | 1 | | BOITE_LATERAL | MDF |
| 4 | 1 | | BOITE_LATERAL_MOTEUR | MDF |
| 3 | 1 | | BOITE_FOND | MDF |
| 2 | 12 | | EQUERRE_INT | X 10 Cr Ni S 18 10 |
| 1 | 1 | | BOITE_AV | MDF |

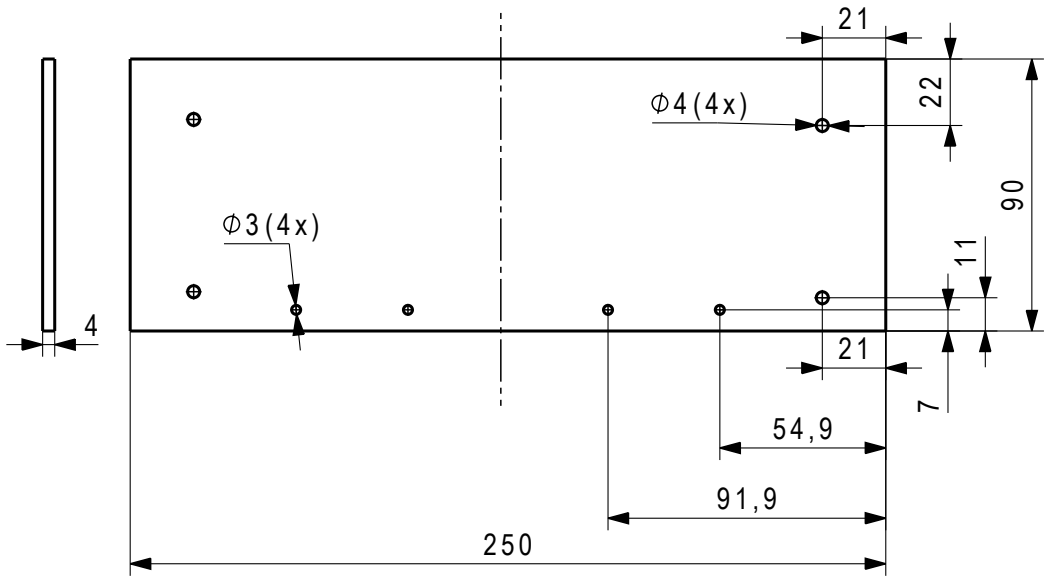
| | | | |
|---|---|---|---|
| Mod. | Mod. | Dessiné 09.01.2021 MAXIME | Echelle 1:5 |
| | | Contrôlé | |
| | | Conf aux norm | |
| | | Bon pour exéc. | |
| Sans nomenclature séparée | | N° de commande | |
| Nomenclature sép de même N° | | Origine | Format A3 / Nb feuilles 1 / Feuille N° 1 |
| Nomenclature sép de N° diff | | Remplace | |
| EPFL | Dénomination BOX | | N° de dessin |


1
2
3
4
A
B
C
D
E
F
Φ4 (2x)
21
22
250
R40
R40
Φ4 (2x)
30
4
21
30
220
250
Laser cut MDF 4mm
Mod.
Mod.
Dessiné
08.01.2021
MAXIME
Echelle
1:2
Contrôlé
Conf aux norm
Bon pour exéc.
Sans nomenclature séparée
Nomenclature sép de même N°
Nomenclature sép de N° diff
N° de commande
Matiere
MDF
Origine
Masse
0,127 kg
Remplace
Format
A4
Nb feuilles
1
Feuille N°
1
EPFL
Dénomination
BOITE_AV
N° de dessin



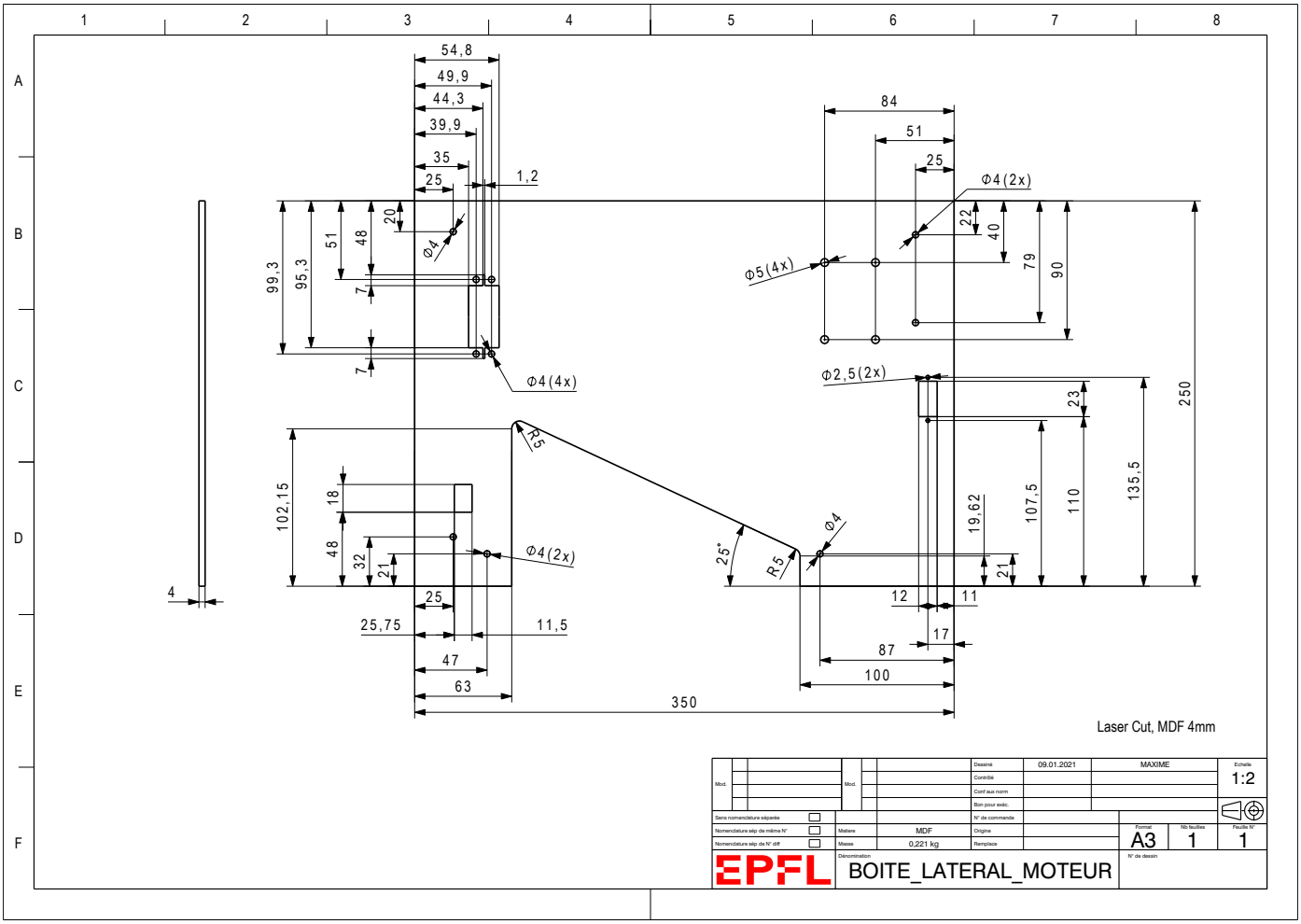

Manufactured, stainless steel

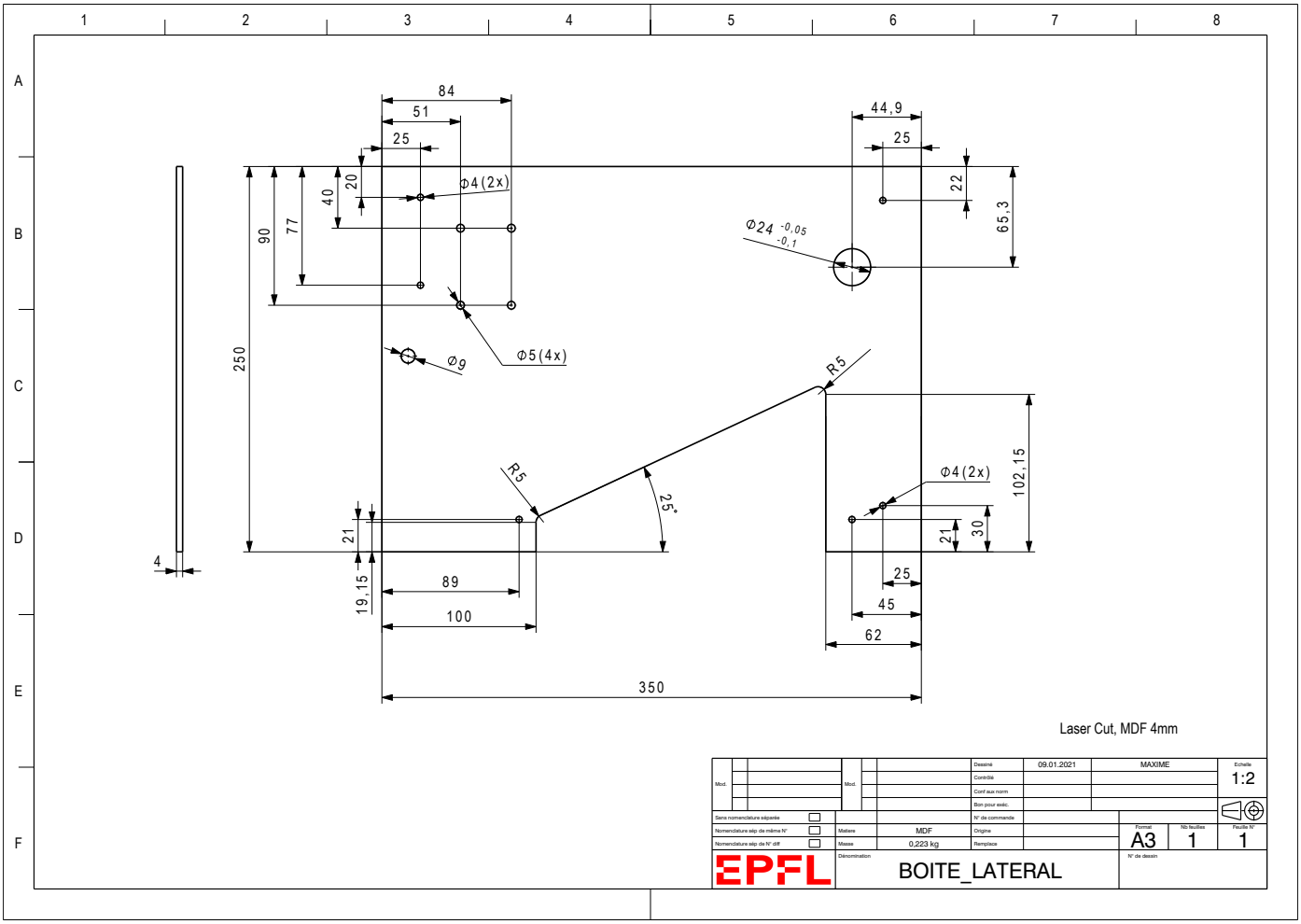
| | | | | | |
|---|---|---|---|---|---|
| Mod. | Mod. | Dessiné | 08.01.2021 | MAXIME | Echelle 1:1 |
| | | Contrôlé | | | |
| | | Conf aux norm | | | |
| | | Bon pour exéc. | | | |
| Sans nomenclature séparée ☐ | | N° de commande | | | |
| Nomenclature sép de même N° ☐ | Matiere: X 10 Cr Ni S 18 10 | Origine | | Format A4 | Nb feuilles 1 / Feuille N° 1 |
| Nomenclature sép de N° diff ☐ | Masse: 0,01 kg | Remplace | | | |
| EPFL | Dénomination: EQUERRE_INT | | | N° de dessin | |

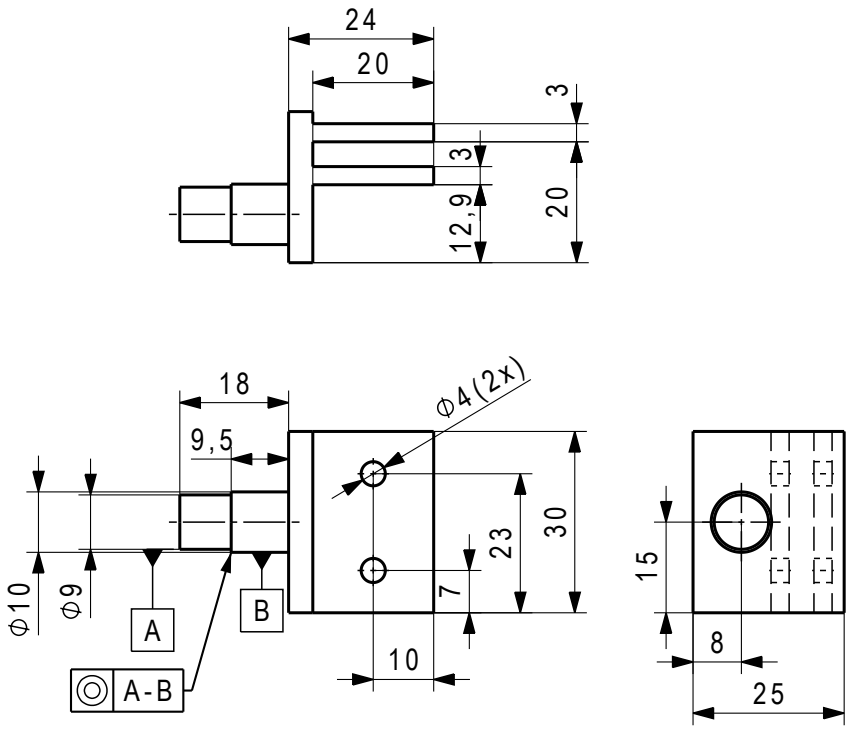



Laser Cut, MDF

| | | | | | |
|---|---|---|---|---|---|
| Mod. | Mod. | Dessiné | 08.01.2021 | MAXIME | Echelle 1:2 |
| | | Contrôlé | | | |
| | | Conf aux norm | | | |
| | | Bon pour exéc. | | | |
| Sans nomenclature séparée ☐ | | N° de commande | | | |
| Nomenclature sép de même N° ☐ | Matiere: MDF | Origine | | Format: A4 | Nb feuilles: 1 / Feuille N°: 1 |
| Nomenclature sép de N° diff ☐ | Masse: 0,067 kg | Remplace | | | |
| EPFL | Dénomination: BOITE_FOND | | | N° de dessin | |

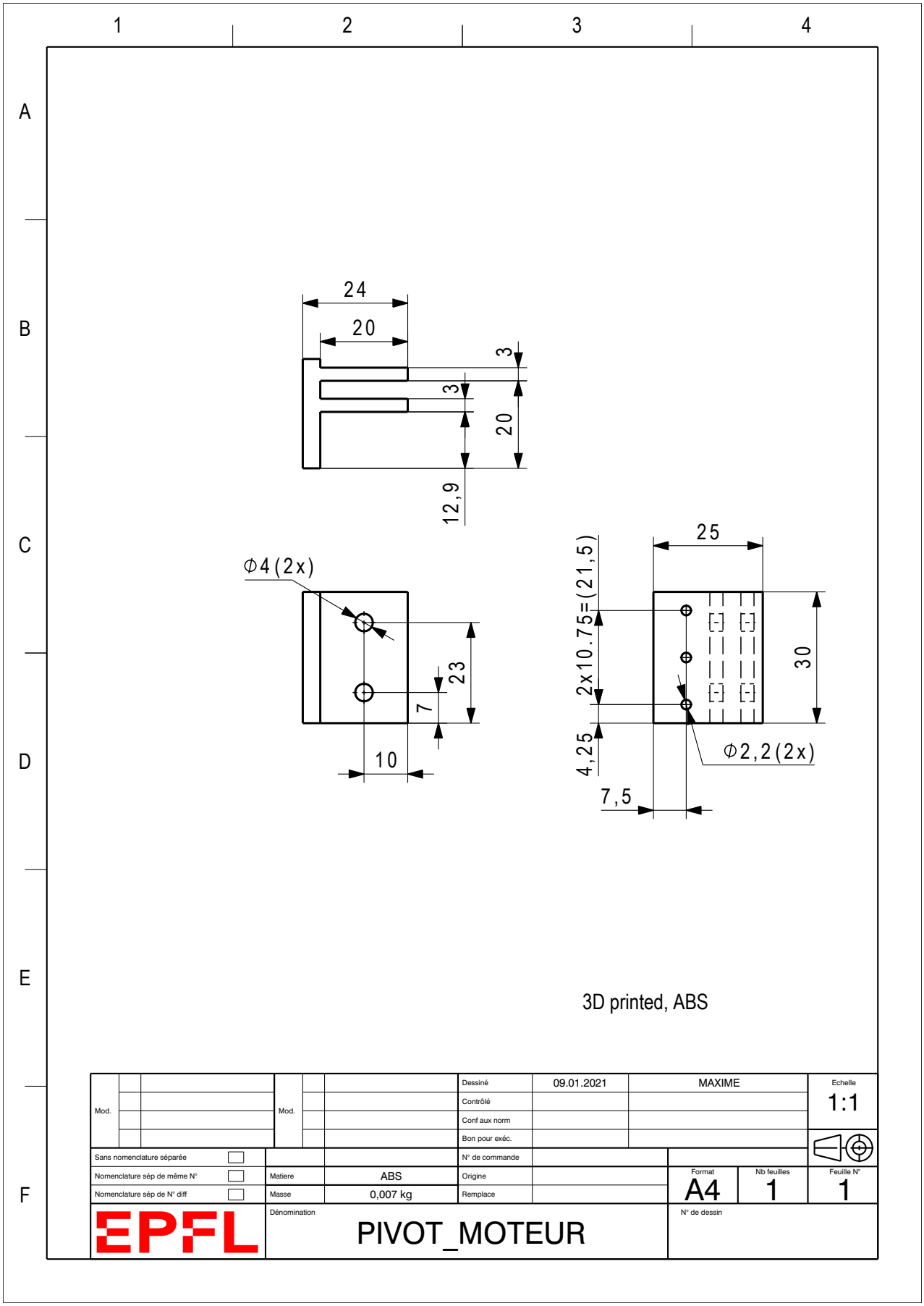

54,8
49,9
44,3
39,9
35
25
1,2
84
51
25
Φ4(2x)
99,3
95,3
51
48
20
Φ4
7
7
Φ4(4x)
Φ5(4x)
22
40
79
90
Φ2,5(2x)
23
250
R5
102,15
18
48
32
21
4
Φ4(2x)
25
25,75
11,5
47
63
350
25°
R5
Φ4
19,62
107,5
110
135,5
21
12
11
17
87
100
Laser Cut, MDF 4mm
Dessiné 09.01.2021 MAXIME
Contrôlé
Conf aux norm
Bon pour exéc.
Echelle 1:2
N° de commande
Matière MDF
Origine
Masse 0,221 kg
Remplace
Format A3
Nb feuilles 1
Feuille N° 1
Dénomination BOITE_LATERAL_MOTEUR
N° de dessin
EPFL

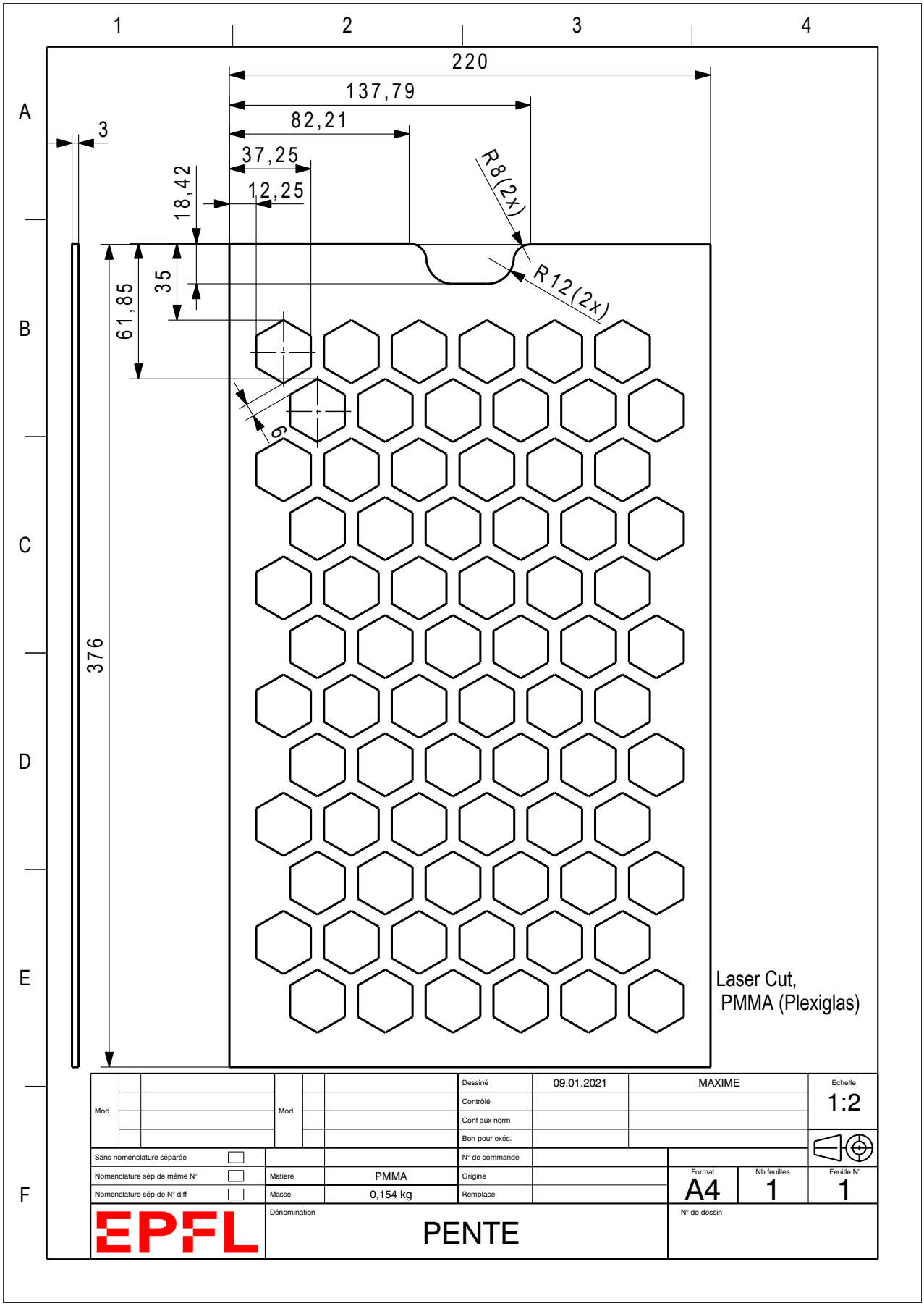

84
51
25
44,9
25
Φ4(2x)
20
40
90
77
22
65,3
Φ24 -0,05 -0,1
Φ9
Φ5(4x)
250
R5
R5
25°
Φ4(2x)
102,15
21
30
21
4
19,15
89
100
25
45
62
350
Laser Cut, MDF 4mm
Dessiné 09.01.2021 MAXIME
Contrôlé
Conf aux norm
Bon pour exéc.
Echelle 1:2
N° de commande
Matière MDF
Origine
Masse 0,223 kg
Remplace
Format A3
Nb feuilles 1
Feuille N° 1
Dénomination BOITE_LATERAL
N° de dessin

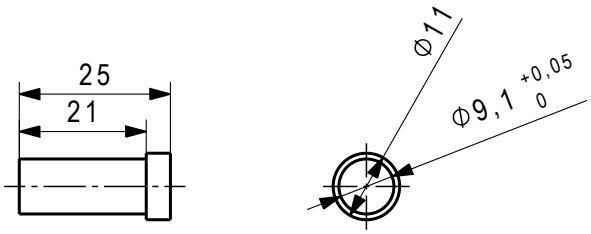



3D printed, ABS

| | | | | | |
|---|---|---|---|---|---|
| Mod. | Mod. | Dessiné | 09.01.2021 | MAXIME | Echelle 1:1 |
| | | Contrôlé | | | |
| | | Conf aux norm | | | |
| | | Bon pour exéc. | | | |
| Sans nomenclature séparée ☐ | | N° de commande | | | |
| Nomenclature sép de même N° ☐ | Matiere: ABS | Origine | | Format A4 | Nb feuilles 1 / Feuille N° 1 |
| Nomenclature sép de N° diff ☐ | Masse: 0,008 kg | Remplace | | | |
| EPFL | Dénomination: PIVOT_ROULEMENT | | | N° de dessin | |

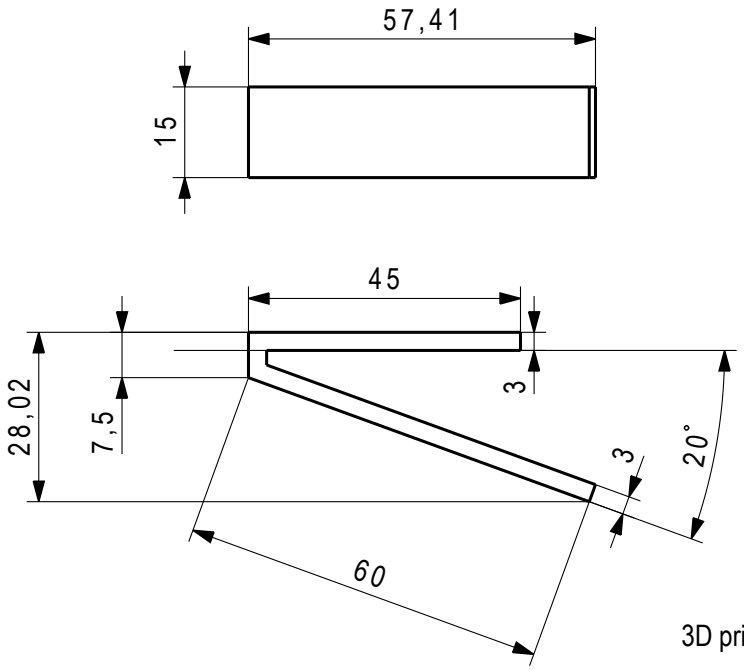

1
2
3
4
A
B
C
D
E
F
24
20
3
3
20
12,9
Φ4 (2x)
23
7
10
25
2x10.75=(21,5)
30
4,25
7,5
Φ2,2 (2x)
3D printed, ABS
Mod.
Mod.
Dessiné
09.01.2021
MAXIME
Contrôlé
Conf aux norm
Bon pour exéc.
Echelle
1:1
Sans nomenclature séparée
Nomenclature sép de même N°
Nomenclature sép de N° diff
N° de commande
Matiere
ABS
Origine
Masse
0,007 kg
Remplace
Format
A4
Nb feuilles
1
Feuille N°
1
EPFL
Dénomination
PIVOT_MOTEUR
N° de dessin


1
2
3
4
A
B
C
D
E
F
220
137,79
82,21
37,25
12,25
18,42
3
61,85
35
376
R8(2x)
R12(2x)
6
Laser Cut,
PMMA (Plexiglas)
Mod.
Mod.
Dessiné
09.01.2021
MAXIME
Contrôlé
Conf aux norm
Bon pour exéc.
Echelle
1:2
Sans nomenclature séparée
Nomenclature sép de même N°
Nomenclature sép de N° diff
N° de commande
Matiere
PMMA
Origine
Masse
0,154 kg
Remplace
Format
A4
Nb feuilles
1
Feuille N°
1
EPFL
Dénomination
PENTE
N° de dessin



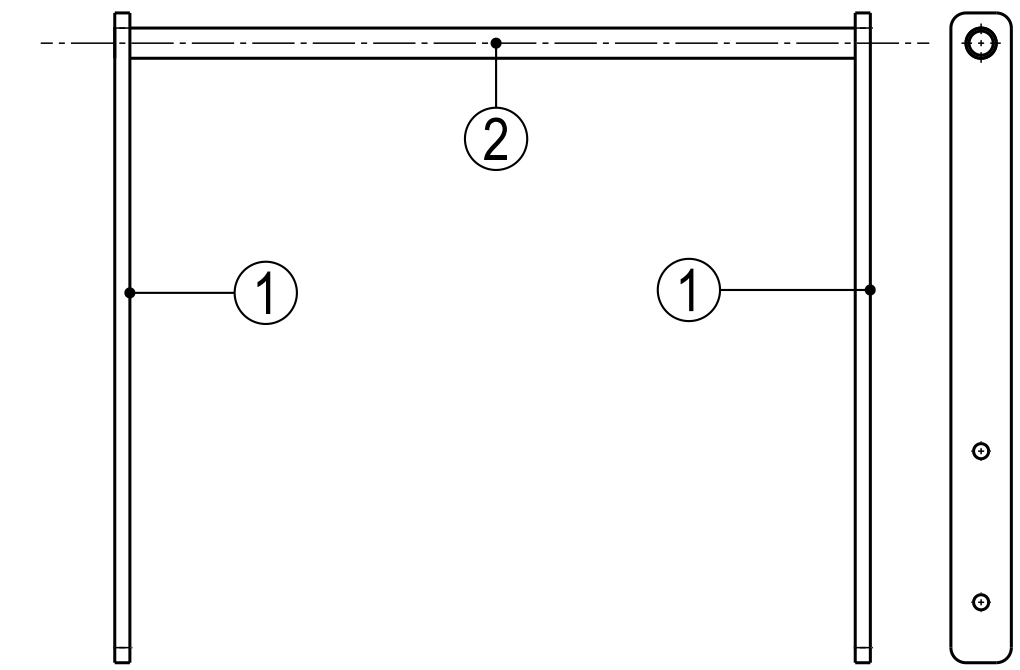

3D printed, ABS

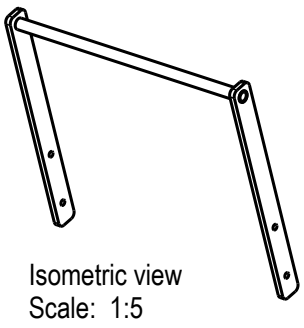
| | | | | | |
|---|---|---|---|---|---|
| Mod. | Mod. | Dessiné | 09.01.2021 | MAXIME | Echelle 1:1 |
| | | Contrôlé | | | |
| | | Conf aux norm | | | |
| | | Bon pour exéc. | | | |
| Sans nomenclature séparée ☐ | | N° de commande | | | |
| Nomenclature sép de même N° ☐ | Matiere ABS | Origine | | Format A4 / Nb feuilles 1 | Feuille N° 1 |
| Nomenclature sép de N° diff ☐ | Masse 0,002 kg | Remplace | | | |
| EPFL | Dénomination AXE_BRAS_ROULEMENT | | | N° de dessin | |




3D printed, ABS

| | | | | | |
|---|---|---|---|---|---|
| Mod. | Dessiné | 09.01.2021 | MAXIME | Echelle | 1:1 |
| Mod. | Contrôlé | | | | |
| | Conf aux norm | | | | |
| | Bon pour exéc. | | | | |
| Sans nomenclature séparée ☐ | N° de commande | | | | |
| Nomenclature sép de même N° ☐ | Matiere | ABS | Origine | Format A4 | Nb feuilles 1 / Feuille N° 1 |
| Nomenclature sép de N° diff ☐ | Masse | 0,005 kg | Remplace | | |
| EPFL | Dénomination | SUPPORT_SENSORS_UP | | N° de dessin | |



Isometric view
Scale: 1:5

| Pos. | Quantité | Unité | Numéro d'identification | Dénomination/caractéristiques | |
|---|---|---|---|---|---|
| 2 | 1 | | LIAISON_BUTTEE_INT | EN AC-AlSi7Mg0.3 T6 | |
| 1 | 2 | | BRAS_BUTTEE_INT | MDF | |

| | | | |
|---|---|---|---|
| Dessiné | 10.01.2021 | MAXIME | Echelle 1:2 |
| Contrôlé | | | |
| Conf aux norm | | | |
| Bon pour exéc. | | | |

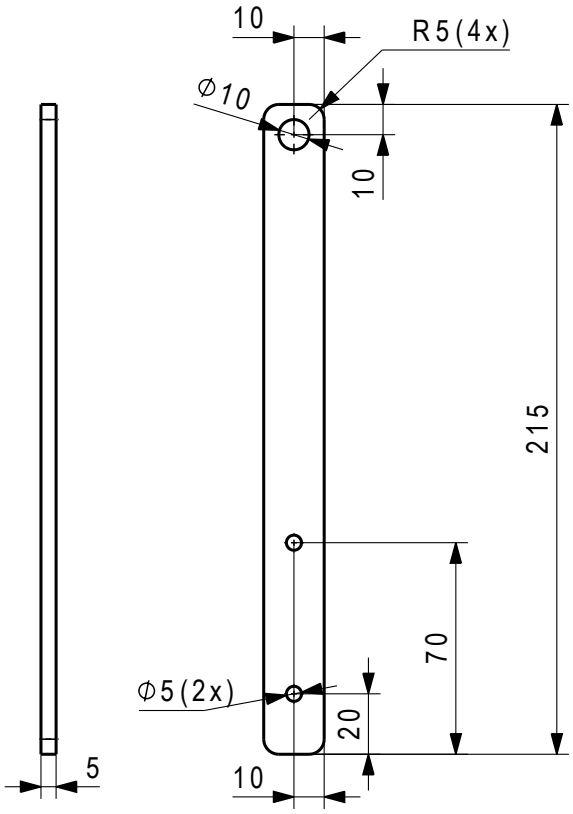
| | | | |
|---|---|---|---|
| Sans nomenclature séparée ☐ | | N° de commande | |
| Nomenclature sép de même N° ☐ | | Origine | |
| Nomenclature sép de N° diff ☐ | | Remplace | |

| Format | Nb feuilles | Feuille N° |
|---|---|---|
| A4 | 1 | 1 |

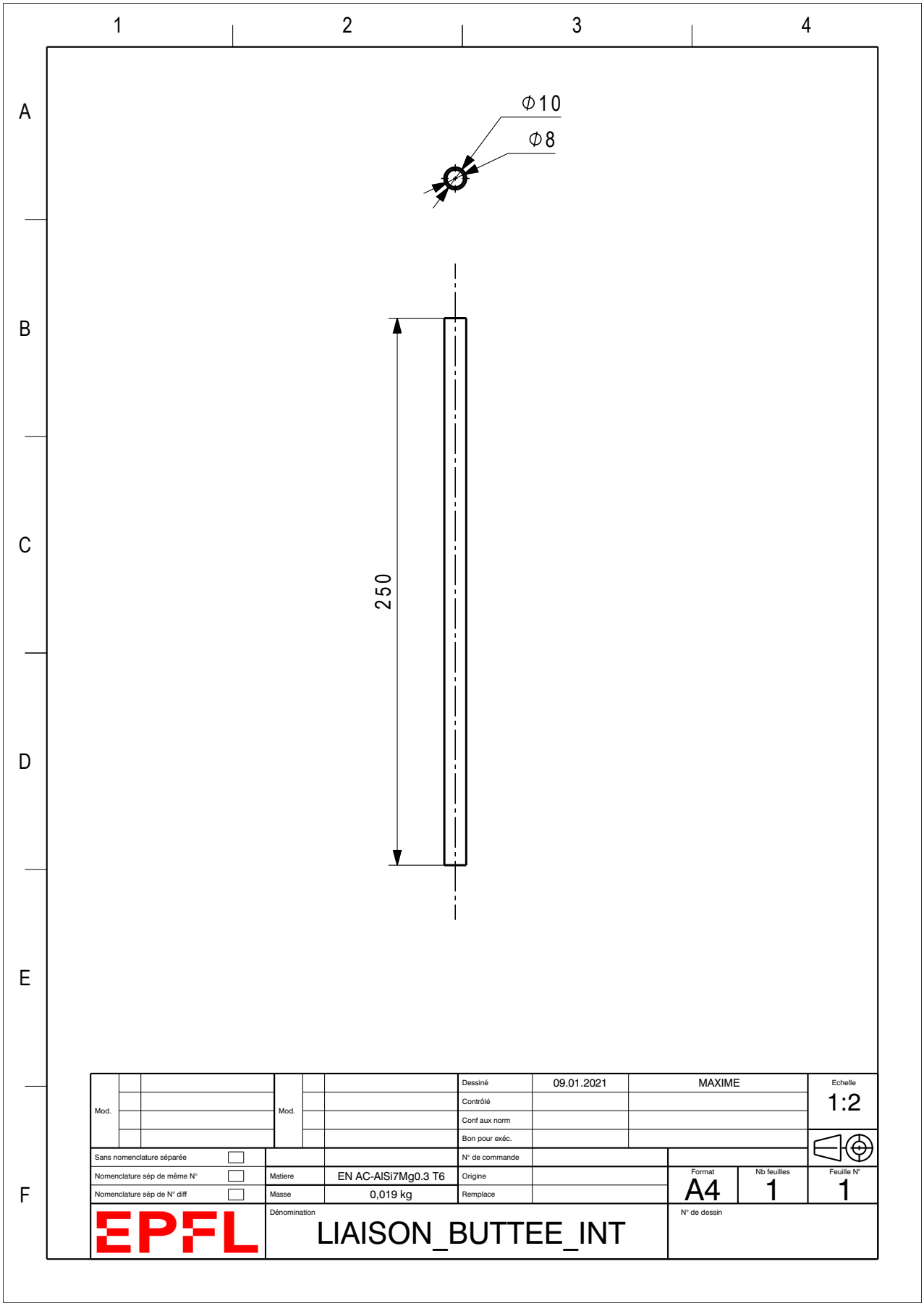
EPFL

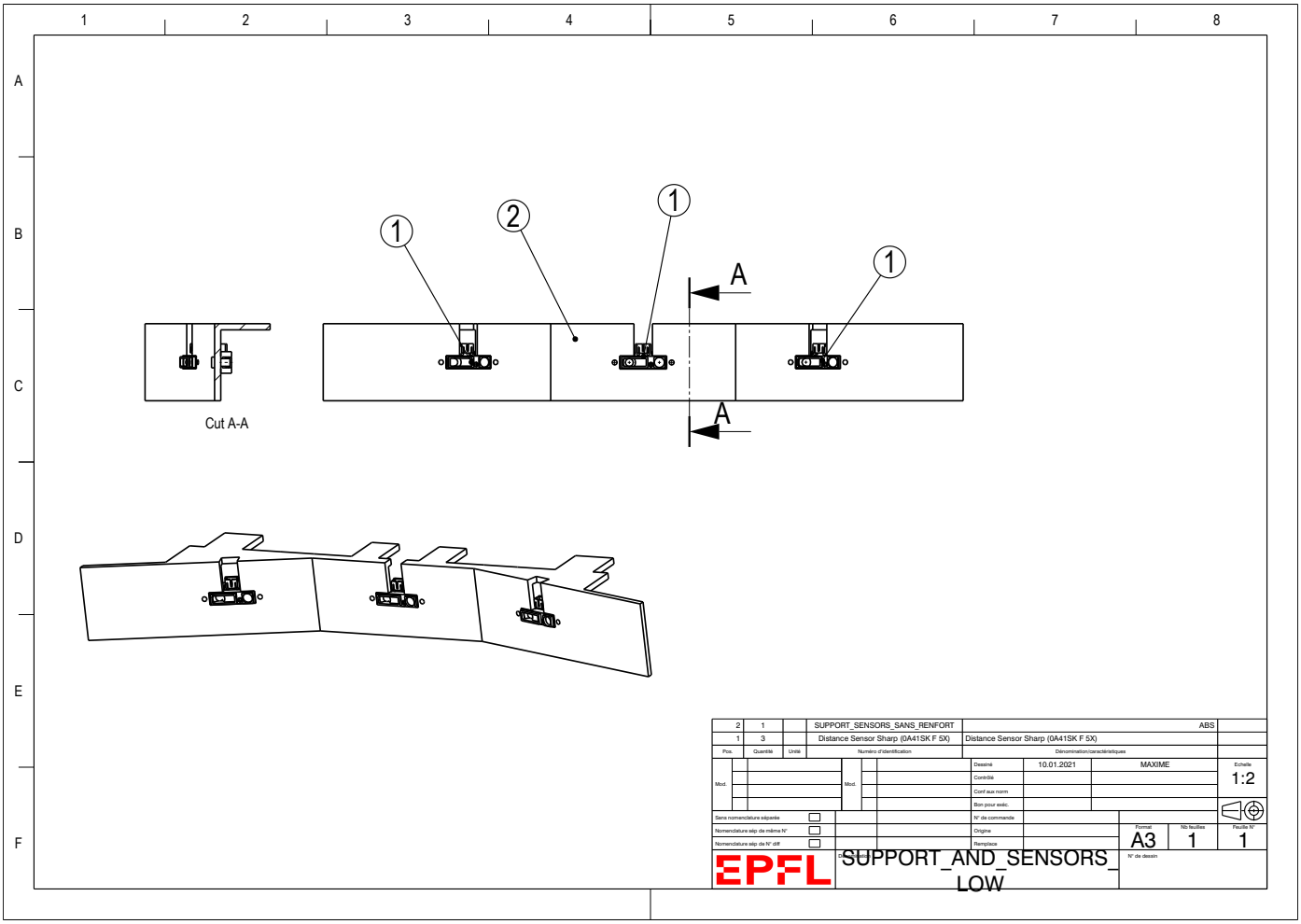
Dénomination: STOP INT

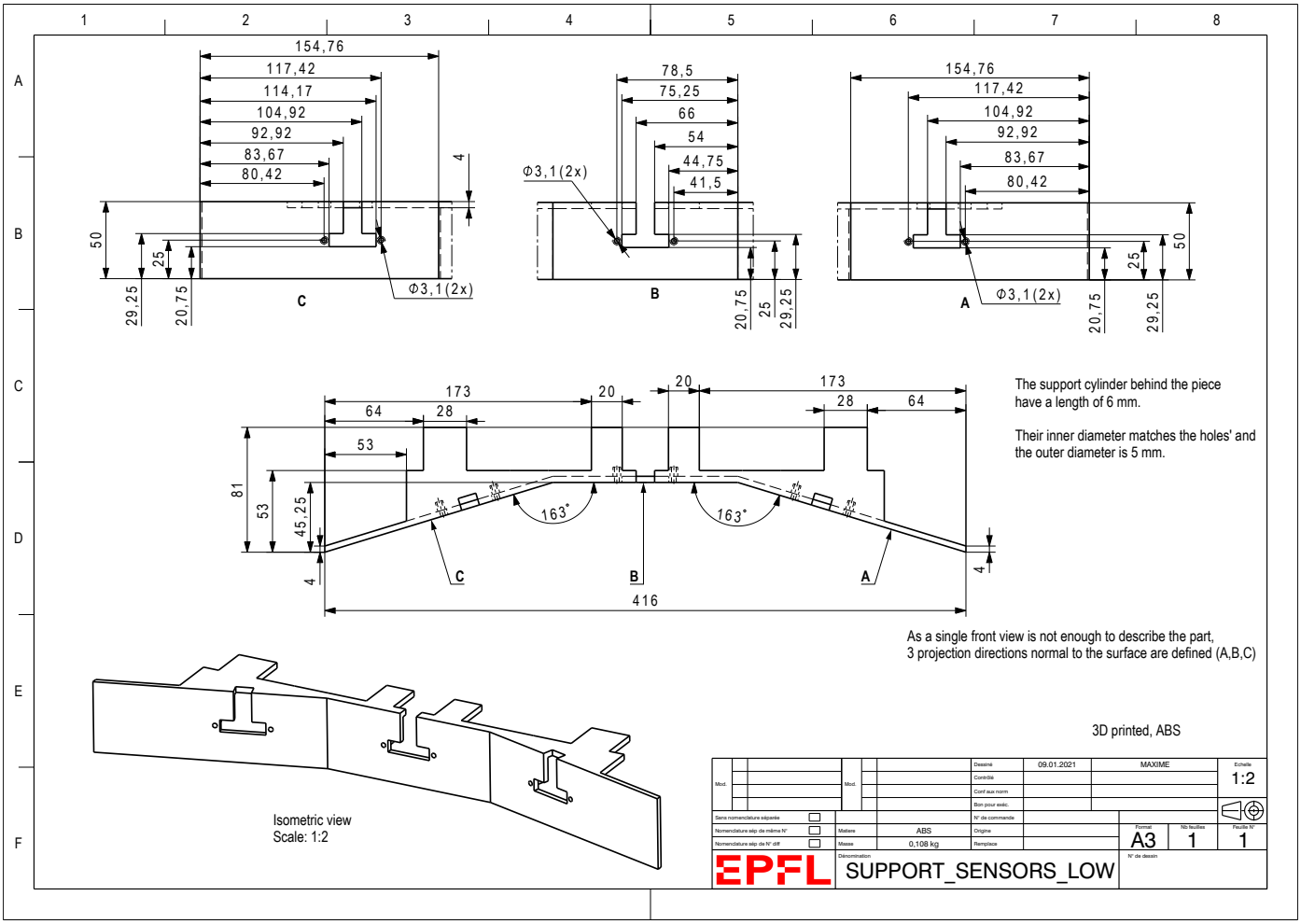
N° de dessin:

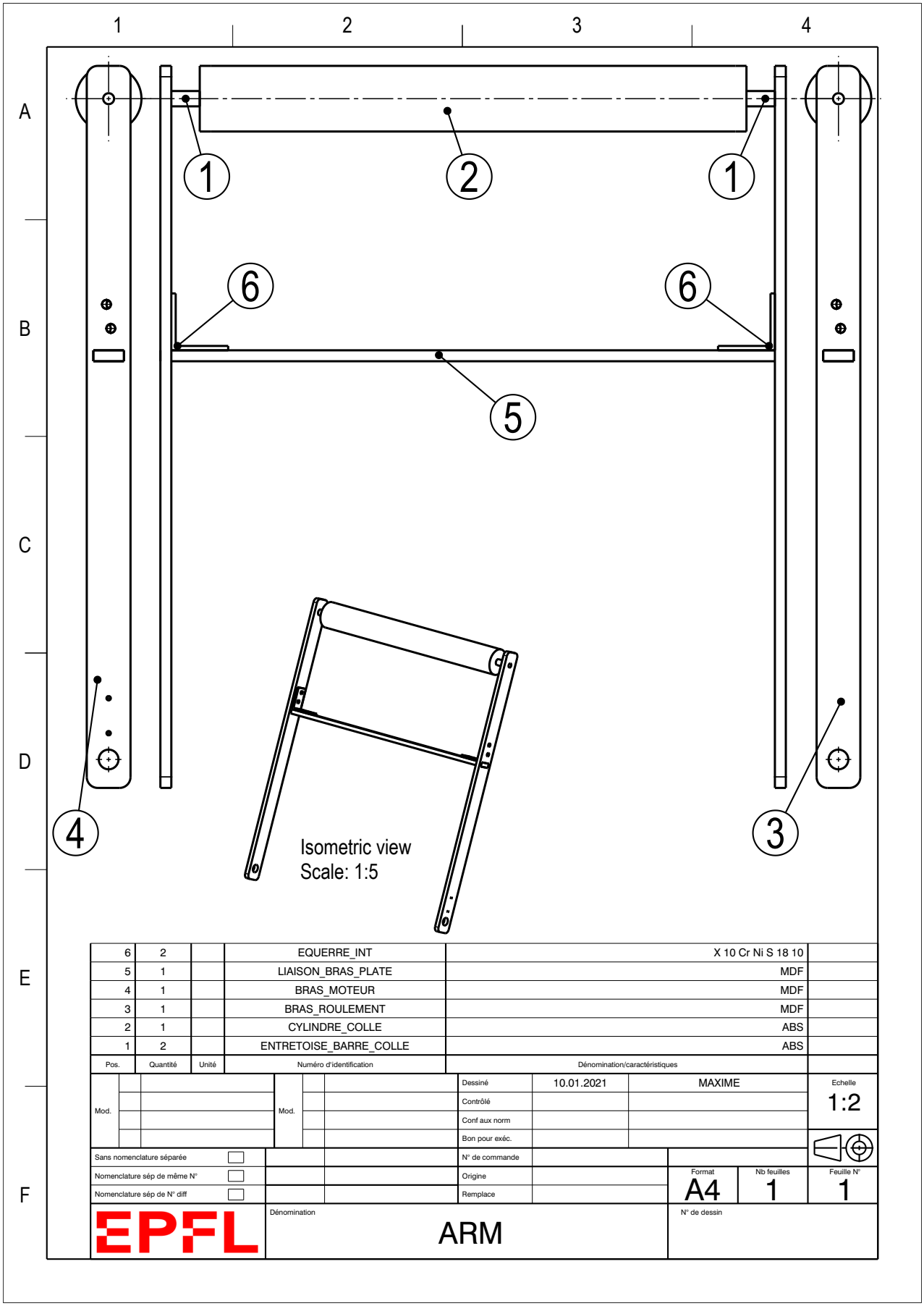



| | | | | | |
|---|---|---|---|---|---|
| Mod. | Mod. | Dessiné | 09.01.2021 | MAXIME | Echelle 1:2 |
| | | Contrôlé | | | |
| | | Conf aux norm | | | |
| | | Bon pour exéc. | | | |
| Sans nomenclature séparée ☐ | | N° de commande | | | |
| Nomenclature sép de même N° ☐ | Matiere: MDF | Origine | | Format: A4 | Nb feuilles: 1 / Feuille N°: 1 |
| Nomenclature sép de N° diff ☐ | Masse: 0,016 kg | Remplace | | | |
| EPFL | Dénomination: BRAS_BUTTEE_INT | | | N° de dessin | |


1
2
3
4
A
B
C
D
E
F
Φ10
Φ8
250
Mod.
Mod.
Dessiné
09.01.2021
MAXIME
Contrôlé
Conf aux norm
Bon pour exéc.
Echelle
1:2
Sans nomenclature séparée
Nomenclature sép de même N°
Nomenclature sép de N° diff
Matiere
EN AC-AlSi7Mg0.3 T6
Masse
0,019 kg
N° de commande
Origine
Remplace
Format
A4
Nb feuilles
1
Feuille N°
1
EPFL
Dénomination
LIAISON_BUTTEE_INT
N° de dessin

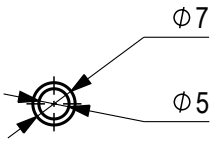

1
2
3
4
5
6
7
8
A
B
C
D
E
F
1
2
1
A
1
A
Cut A-A
2 1 SUPPORT_SENSORS_SANS_RENFORT ABS
1 3 Distance Sensor Sharp (0A41SK F 5X) Distance Sensor Sharp (0A41SK F 5X)
Pos. Quantité Unité Numéro d'identification Dénomination/caractéristiques
Mod. Mod.
Dessiné 10.01.2021 MAXIME
Contrôlé
Conf aux norm
Bon pour exéc.
Echelle 1:2
Sans nomenclature séparée
Nomenclature sép de même N°
Nomenclature sép de N° diff
N° de commande
Origine
Remplace
Format A3
Nb feuilles 1
Feuille N° 1
N° de dessin
EPFL
SUPPORT_AND_SENSORS_LOW

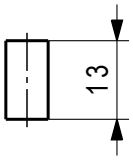

154,76
117,42
114,17
104,92
92,92
83,67
80,42
4
50
25
29,25
20,75
C
Φ3,1(2x)
78,5
75,25
66
54
44,75
41,5
Φ3,1(2x)
20,75
25
29,25
B
154,76
117,42
104,92
92,92
83,67
80,42
Φ3,1(2x)
25
50
20,75
29,25
A
173
64
28
20
20
173
28
64
53
81
53
45,25
163°
163°
4
C
B
A
4
416
The support cylinder behind the piece have a length of 6 mm.
Their inner diameter matches the holes' and the outer diameter is 5 mm.
As a single front view is not enough to describe the part, 3 projection directions normal to the surface are defined (A,B,C)
Isometric view
Scale: 1:2
3D printed, ABS
Dessiné 09.01.2021 MAXIME
Contrôlé
Conf aux norm
Bon pour exéc.
N° de commande
Origine
Remplace
Matière ABS
Masse 0,108 kg
Dénomination
SUPPORT_SENSORS_LOW
Format A3
Nb feuilles 1
Feuille N° 1
Echelle 1:2
N° de dessin




| Pos. | Quantité | Unité | Numéro d'identification | Dénomination/caractéristiques | |
|---|---|---|---|---|---|
| 6 | 2 | | EQUERRE_INT | X 10 Cr Ni S 18 10 | |
| 5 | 1 | | LIAISON_BRAS_PLATE | MDF | |
| 4 | 1 | | BRAS_MOTEUR | MDF | |
| 3 | 1 | | BRAS_ROULEMENT | MDF | |
| 2 | 1 | | CYLINDRE_COLLE | ABS | |
| 1 | 2 | | ENTRETOISE_BARRE_COLLE | ABS | |

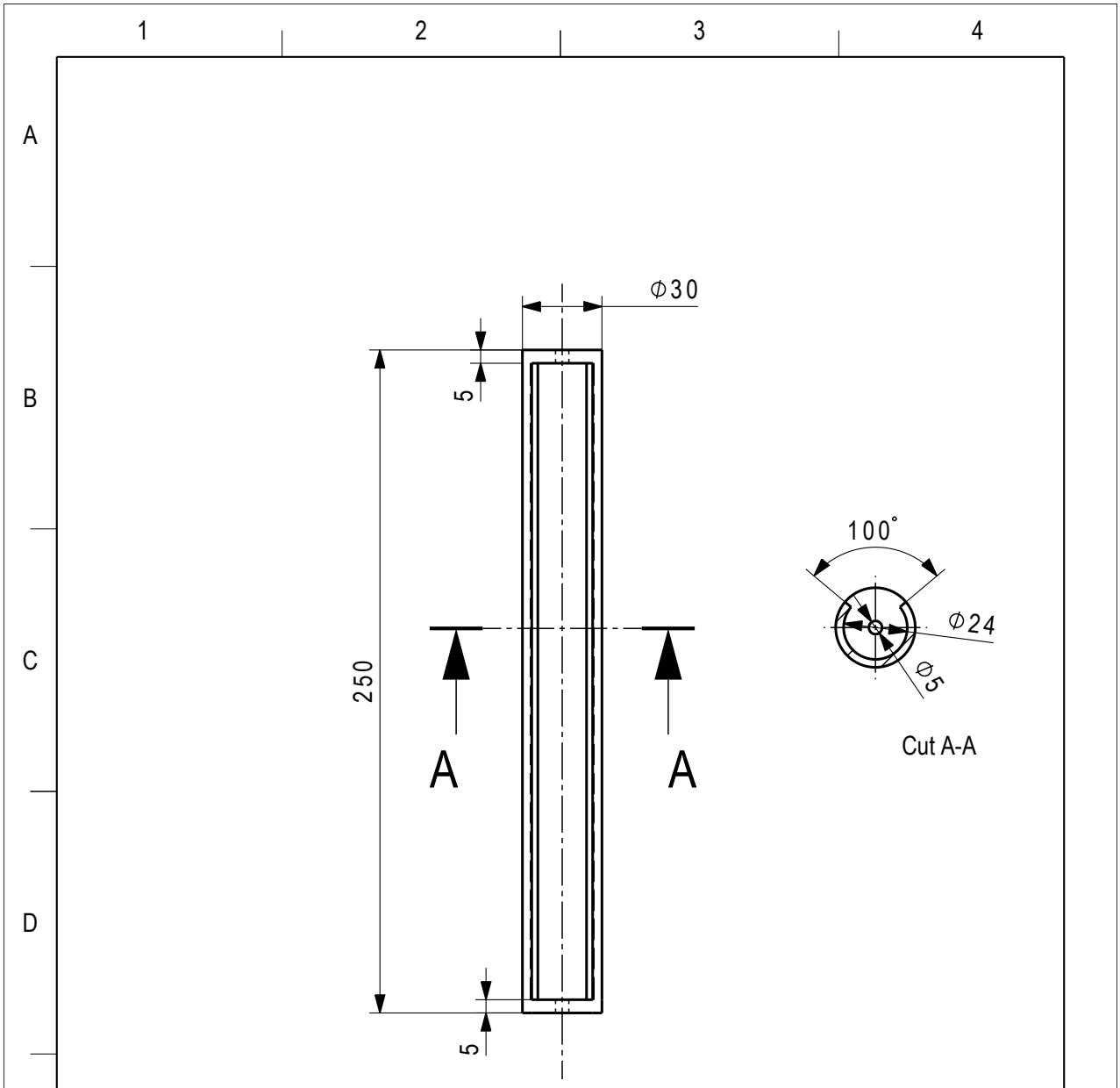
| | | | | | |
|---|---|---|---|---|---|
| Mod. | Mod. | Dessiné | 10.01.2021 | MAXIME | Echelle 1:2 |
| | | Contrôlé | | | |
| | | Conf aux norm | | | |
| | | Bon pour exéc. | | | |
| Sans nomenclature séparée ☐ | | N° de commande | | | |
| Nomenclature sép de même N° ☐ | | Origine | | Format A4 / Nb feuilles 1 | Feuille N° 1 |
| Nomenclature sép de N° diff ☐ | | Remplace | | | |
| EPFL | Dénomination: ARM | | | N° de dessin | |



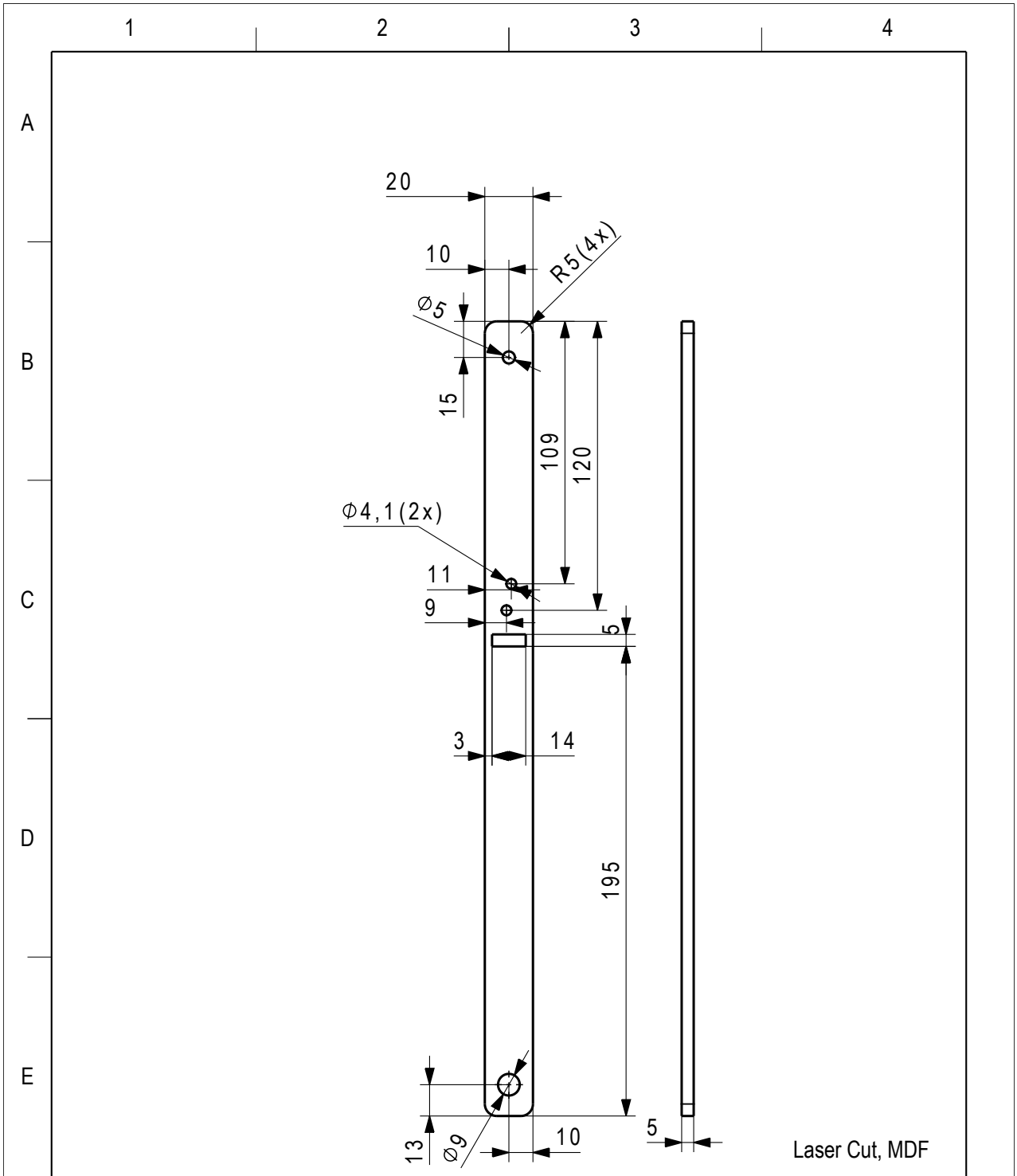



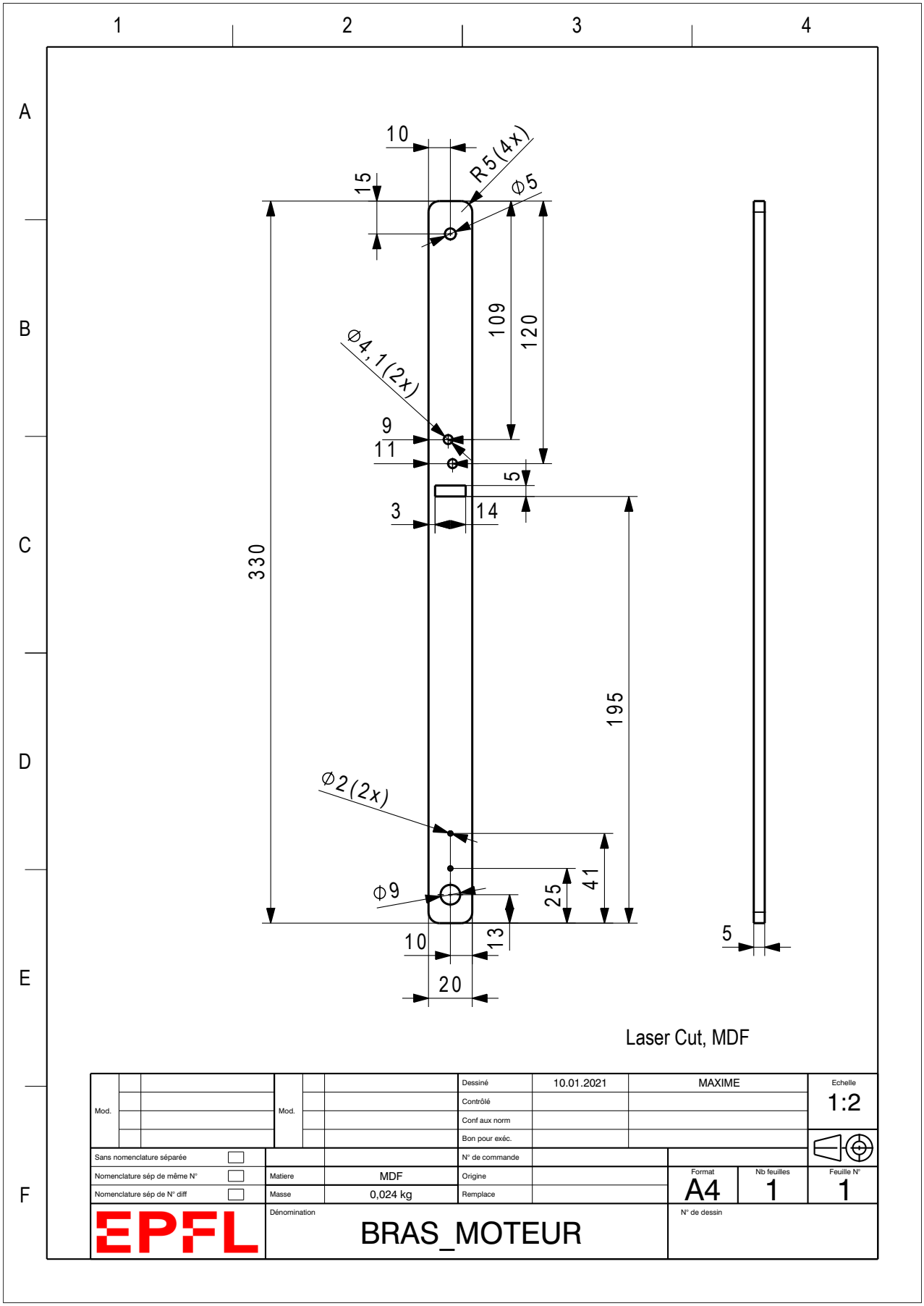

3D printed, ABS

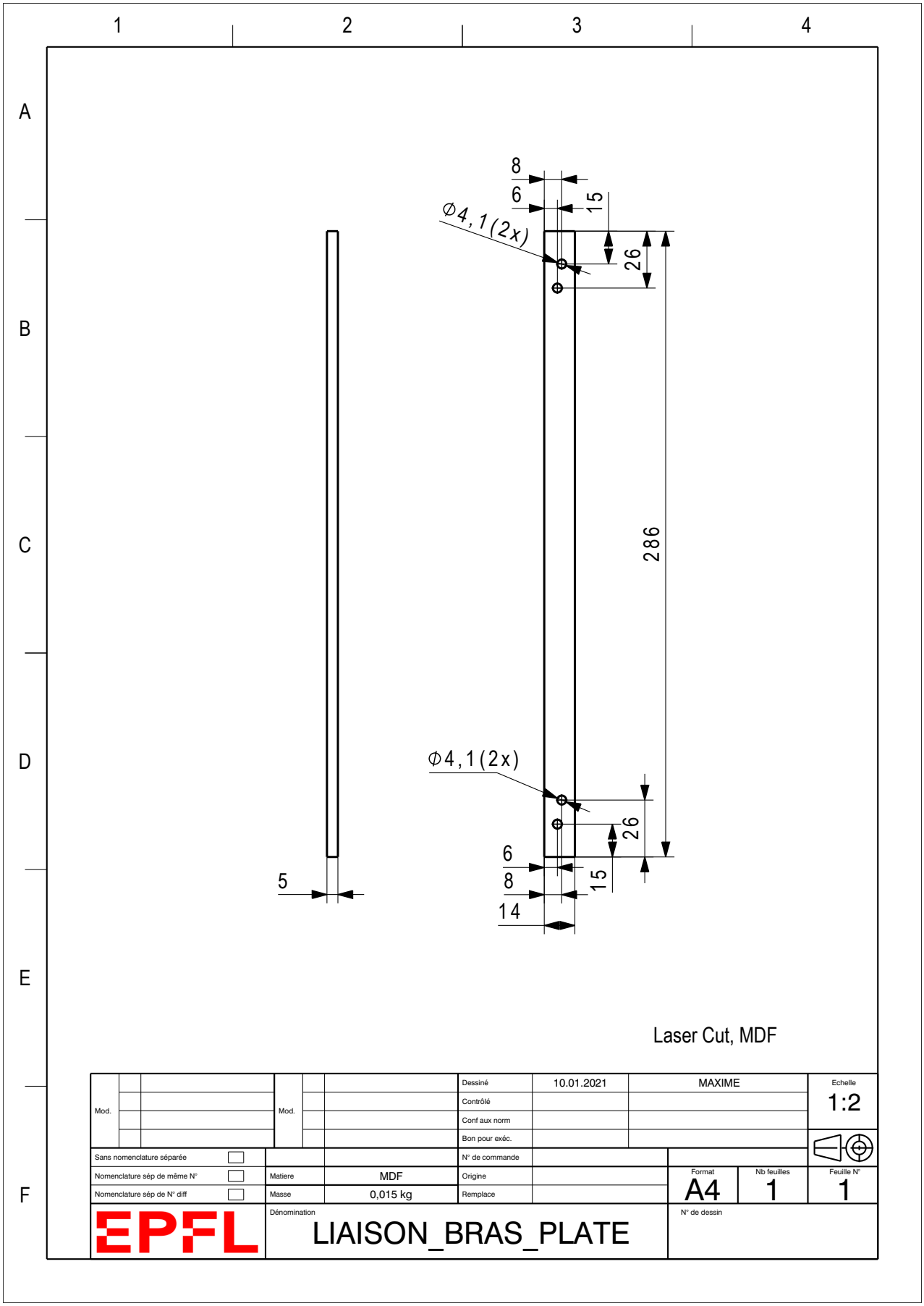
| | | | | | |
|---|---|---|---|---|---|
| Mod. | Mod. | Dessiné | 10.01.2021 | MAXIME | Echelle 1:1 |
| | | Contrôlé | | | |
| | | Conf aux norm | | | |
| | | Bon pour exéc. | | | |
| Sans nomenclature séparée ☐ | | N° de commande | | | |
| Nomenclature sép de même N° ☐ | Matiere ABS | Origine | | Format A4 / Nb feuilles 1 | Feuille N° 1 |
| Nomenclature sép de N° diff ☐ | Masse 0 kg | Remplace | | | |
| EPFL | Designation ENTRETOISE_BARRE_COLLE | | | N° de dessin | |

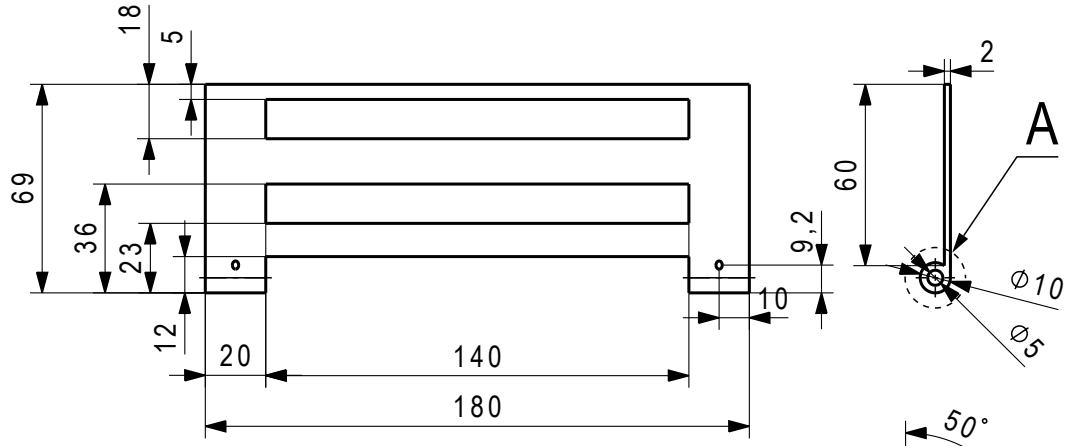


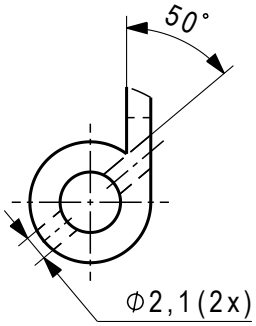

| | | | | | |
|---|---|---|---|---|---|
| Mod. | Mod. | Dessiné | 10.01.2021 | MAXIME | Echelle 1:2 |
| | | Contrôlé | | | |
| | | Conf aux norm | | | |
| | | Bon pour exéc. | | | |
| Sans nomenclature séparée ☐ | | N° de commande | | | |
| Nomenclature sép de même N° ☐ | Matiere: ABS | Origine | | Format A4 | Nb feuilles 1 / Feuille N° 1 |
| Nomenclature sép de N° diff ☐ | Masse: 0,053 kg | Remplace | | | |
| EPFL | Dénomination: CYLINDRE_COLLE | | | N° de dessin | |


1
2
3
4
A
B
C
D
E
F
20
10
R5(4x)
Ø5
15
109
120
Ø4,1(2x)
11
9
5
3
14
195
13
Ø9
10
5
Laser Cut, MDF
Mod.
Mod.
Dessiné
10.01.2021
MAXIME
Echelle
1:2
Contrôlé
Conf aux norm
Bon pour exéc.
Sans nomenclature séparée
Nomenclature sép de même N°
Nomenclature sép de N° diff
N° de commande
Matiere
MDF
Origine
Masse
0,024 kg
Remplace
Format
A4
Nb feuilles
1
Feuille N°
1
EPFL
Dénomination
BRAS_ROULEMENT
N° de dessin


1
2
3
4
A
B
C
D
E
F
10
15
R5(4x)
Ø5
109
120
Ø4,1(2x)
9
11
5
3
14
330
195
Ø2(2x)
Ø9
25
41
10
13
20
5
Laser Cut, MDF
Mod.
Mod.
Dessiné
10.01.2021
MAXIME
Contrôlé
Conf aux norm
Bon pour exéc.
Echelle
1:2
Sans nomenclature séparée
Nomenclature sép de même N°
Nomenclature sép de N° diff
N° de commande
Matiere
MDF
Origine
Masse
0,024 kg
Remplace
Format
A4
Nb feuilles
1
Feuille N°
1
EPFL
Dénomination
BRAS_MOTEUR
N° de dessin


1
2
3
4
A
B
C
D
E
F
8
6
15
Ø4,1(2x)
26
286
Ø4,1(2x)
26
6
8
15
14
5
Laser Cut, MDF
Mod.
Mod.
Dessiné
10.01.2021
MAXIME
Echelle
1:2
Contrôlé
Conf aux norm
Bon pour exéc.
Sans nomenclature séparée
Nomenclature sép de même N°
Nomenclature sép de N° diff
N° de commande
Matiere
MDF
Origine
Masse
0,015 kg
Remplace
Format
A4
Nb feuilles
1
Feuille N°
1
EPFL
Dénomination
LIAISON_BRAS_PLATE
N° de dessin




Detail A
Scale: 2:1

3D printed, ABS

| | | | | | |
|---|---|---|---|---|---|
| Mod. | Dessiné | 10.01.2021 | MAXIME | Echelle | 1:2 |
| Mod. | Contrôlé | | | | |
| | Conf aux norm | | | | |
| | Bon pour exéc. | | | | |
| Sans nomenclature séparée ☐ | N° de commande | | | | |
| Nomenclature sép de même N° ☐ | Matiere | ABS | Origine | Format A4 | Nb feuilles 1 / Feuille N° 1 |
| Nomenclature sép de N° diff ☐ | Masse | 0,015 kg | Remplace | | |
| EPFL | Dénomination | STOP TOP | | N° de dessin | |

# D Raw budgets

Table 14: Raw virtual budget with all expenses

| Description | Amount |
|---|---|
| Initial budget | 2000.00 |
| Arduino Mega2560 R3 [#11] | -44.50 |
| Return: Arduino Mega2560 R3 [#11] | 44.50 |
| Arduino Mega2560 R3 [#11] | -44.50 |
| Tower Pro MG995R digi hi torque [#1] | -10.00 |
| WildTumper 120x60 mm wheel, 4 mm shaft [#4] | -7.50 |
| Raspberry Pi Camera v2.1 [#12] | -31.55 |
| 34:1 DC motor with encoder [#5] | -36.95 |
| WildTumper 120x60 mm wheel, 4 mm shaft [#10] | -7.50 |
| WildTumper 120x60 mm wheel, 4 mm shaft [#11] | -7.50 |
| WildTumper 120x60 mm wheel, 4 mm shaft [#7] | -7.50 |
| Raspberry Pi 4 (4 GB) [#6] | -58.25 |
| 34:1 DC motor with encoder [#8] | -36.95 |
| 30 cm IR proximity sensor [#9] | -11.95 |
| 30 cm IR proximity sensor [#3] | -11.95 |
| 30 cm IR proximity sensor [#8] | -11.95 |
| 30 cm IR proximity sensor [#11] | -11.95 |
| Tower Pro MG90S micro servo [#3] | -5.00 |
| DFRobot DRI0018 2x15A motor driver [#6] | -45.00 |
| Pixy2 camera pan-tilt support | -25.00 |
| DFRobot DRI0018 2x15A motor driver [#4] | -45.00 |
| Pixy2 camera | -60.00 |
| Digital compass module (boxtec 47024) [#11] | -23.55 |
| Return: 34:1 DC motor with encoder [#5] | 36.95 |
| Return: 34:1 DC motor with encoder [#8] | 36.95 |
| 75:1 DC motor with encoder (HP) [#2] | -36.95 |
| 75:1 DC motor with encoder (HP) [#3] | -36.95 |
| 80 cm IR proximity sensor [#39] | -10.95 |
| 80 cm IR proximity sensor [#26] | -10.95 |
| 80 cm IR proximity sensor [#21] | -10.95 |
| 80 cm IR proximity sensor [#42] | -10.95 |
| 80 cm IR proximity sensor [#43] | -10.95 |
| 80 cm IR proximity sensor [#6] | -10.95 |
| 80 cm IR proximity sensor [#8] | -10.95 |
| 80 cm IR proximity sensor [#36] | -10.95 |
| Ultrasonic range module (HC-SR04) - 11mm spacing [#2] | -4.00 |
| Ultrasonic range module (HC-SR04) - 11mm spacing [#13] | -4.00 |
| micro servo 2.8kg HK15148B [#9] | -5.00 |
| Digital compass module (boxtec 47024) [#5] | -23.55 |
| 9 Degrees of Freedom IMU-9150 [#7] | -8.00 |
| NiMH rechargeable battery pack (7.2 V, 3000 mAh) [#5A] | -19.95 |
| micro servo 2.8kg HK15148B [#5] | -5.00 |
| Return: Pixy2 camera pan-tilt support | 25.00 |
| Return: Raspberry Pi Camera v2.1 [#12] | 31.55 |
| Return: Raspberry Pi 4 (4 GB) [#6] | 58.25 |
| 75:1 DC motor with encoder (HP) [#16] | -36.95 |
| Return: Ultrasonic range module (HC-SR04) - 11mm spacing [#2] | 4.00 |
| Return: Ultrasonic range module (HC-SR04) - 11mm spacing [#13] | 4.00 |
| Digital compass module (boxtec 47024) [#2] | -23.55 |
| Return: micro servo 2.8kg HK15148B [#9] | 5.00 |
| 3D printing: 631.39 $cm^3$ model, 178.98 $cm^3$ support | -275.52 |
| Current budget balance | 1135.13 CHF |
| Total expense | 864.87 CHF |

Table 15: Raw 3D printing detail

| Name | Model ($cm^3$) | Support ($cm^3$) | Price |
|---|---|---|---|
| Pack_BARRE_COLLE | 16.82 | 6.41 | 7.90 CHF |
| Pack_CYLINDRE_COLLE_groupe_3 | 44.59 | 14.44 | 20.07 CHF |
| Pack_PALIER_MOTEUR_INF_Group3 | 71.66 | 5.46 | 26.22 CHF |
| Pack_PALIER_MOTEUR_AV | 10.06 | 3.42 | 4.58 CHF |
| Pack_PALIER_MOTEUR_AV_group3 | 10.07 | 3.42 | 4.59 CHF |
| Pack_PALIER_MOTEUR_AV | 19.96 | 6.54 | 9.01 CHF |
| Pack_CYLINDRE_MOTEUR | 0.00 | 1.63 | 0.55 CHF |
| Pack_AXE_ROUE | 9.23 | 5.12 | 4.88 CHF |
| Pack_PIVOTS_PORTE_GROUPE_3 | 12.38 | 5.07 | 5.93 CHF |
| Pack_PALIER_MOTEUR_ROULEMENT_GROUP_3 | 40.48 | 9.66 | 17.05 CHF |
| Pack_PIVOT_MOTEUR | 12.39 | 5.24 | 5.99 CHF |
| Pack_PIVOT_PORTE_GROUP3 | 13.48 | 5.27 | 6.38 CHF |
| SUPPORT_PB | 24.07 | 12.64 | 12.48 CHF |
| Pack_SUPPORT_SENSORS_LEFT | 70.96 | 12.71 | 28.45 CHF |
| SUPPORT_SENSORS_LEFT | 38.41 | 7.30 | 15.54 CHF |
| SUPPORT_SENSORS_RIGHT_GROUP3 | 0.88 | 4.67 | 1.89 CHF |
| SUPPORT_SENSORS_RIGHT | 38.57 | 7.29 | 15.59 CHF |
| SUPPORT_BOUSSOLE | 8.02 | 4.82 | 4.36 CHF |
| Pack_SUPPORT_SENSOR_AV | 9.32 | 3.32 | 4.30 CHF |
| SUPPORT_SENSORS_AVANT | 8.97 | 1.92 | 3.70 CHF |
| SUPPORT_SENSORS_SANS_RENFORT_RIGHT | 14.82 | 6.53 | 7.26 CHF |
| SUPPORT_SENSORS_AVANT | 47.30 | 11.72 | 20.07 CHF |
| SUPPORT_SENSORS_LEFT | 47.31 | 11.46 | 19.98 CHF |
| SUR_BUTTEE | 2.58 | 0.64 | 1.10 CHF |
| SUR_BUTTEE | 1.41 | 0.65 | 0.70 CHF |
| SUR_BUTTEE | 1.50 | 0.65 | 0.73 CHF |
| Pack_SUR_BUTTEE | 1.30 | 0.50 | 0.61 CHF |
| SUR_BUTTEE | 1.50 | 0.65 | 0.73 CHF |
| SUR_BUTTEE | 1.49 | 0.65 | 0.73 CHF |
| SUR_BUTTEE | 1.50 | 0.65 | 0.73 CHF |
| SUR_BUTTEE | 10.84 | 1.86 | 4.32 CHF |
| SUR_BUTTEE | 1.49 | 0.65 | 0.73 CHF |
| SUR_BUTTEE | 7.15 | 1.87 | 3.07 CHF |
| Pack_SUR_BUTTEE | 15.73 | 3.55 | 6.55 CHF |
| SUR_BUTTEE_1 | 15.15 | 10.58 | 8.75 CHF |
| Total | - | - | 275.52 CHF |

Table 16: Raw real budget

| **Description** | **Amount** |
|---|---|
| Initial budget | 750.00 |
| Impression PETG | -2.10 |
| 2 plaques MDF 4mm + 1 plaque plexi 3mm | -11.40 |
| Commande Distrelec du 17.11.2020 | -50.15 |
| Servo grillé | -5.00 |
| Current balance | 681.35 |
| Total | 68.65 CHF |