

1<sup>ere</sup> STI2D

# ROBOT ASPIRATEUR

DEGUISNE ARIELLE  
NOIRTIN FELIX  
DESCHASAUX GABRIEL  
GAVOILLE THIBAUT

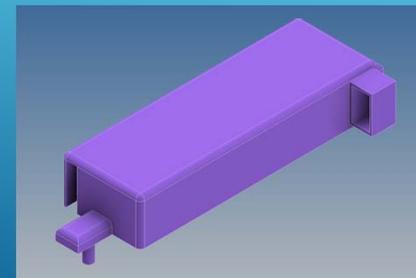
## SOMMAIRE

- 1) Partie : Alimentation du robot aspirateur, surveillance et voyant
- 2) Partie : Capteur de détection d'obstacle à distance et alerte sonore
- 3) Partie : Système d'aspiration du robot aspirateur
- 4) Partie : Déplacement et Bouton Poussoir

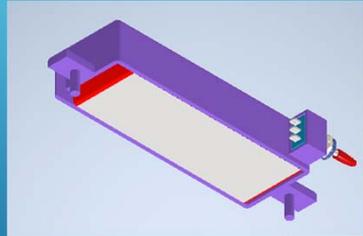
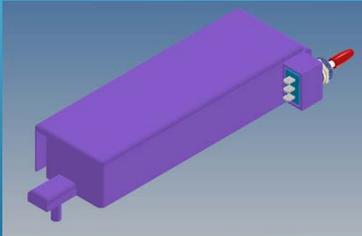
## PARTIE : ALIMENTATION DU ROBOT ASPIRATEUR



## CONCEPTION DU BOITIER DE LA BATTERIE



## ASSEMBLAGE DU BOITIER AVEC LA BATTERIE



## ALGORITHME DE LA BATTERIE

SI Interrupteur ON :

Alors Batterie = ON

Robot = AVANCE

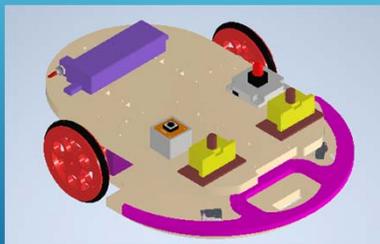
SINON SI Interrupteur OFF :

Alors Batterie = OFF

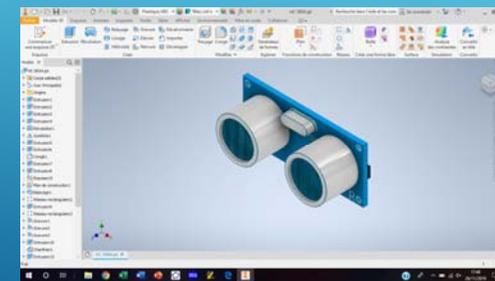
Robot = ARRET

FIN

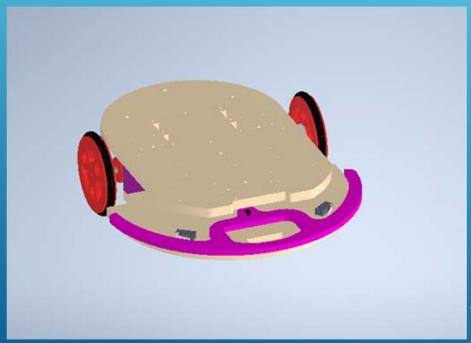
## MISE EN PLACE DE LA BATTERIE



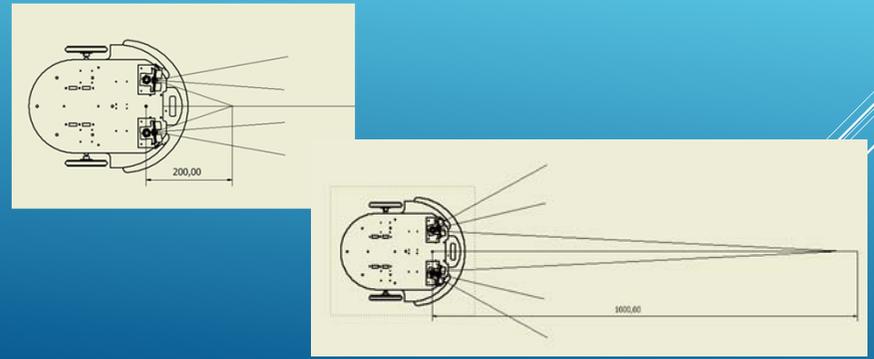
## PARTIE : CAPTEUR DE DÉTECTION D'OBSTACLE À DISTANCE



### LA MAQUETTE: EMBLACEMENT DES CAPTEURS US

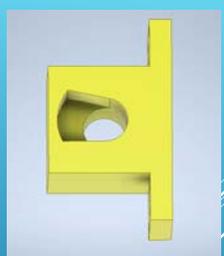
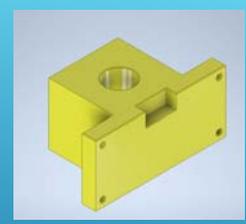
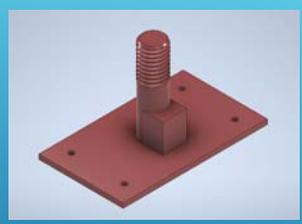


### DISTANCE DE DÉTECTION:

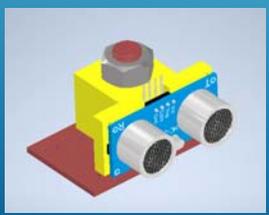


Socle

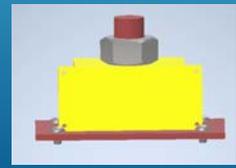
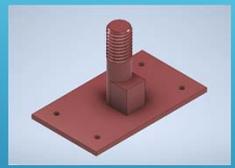
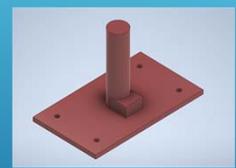
Support du capteur



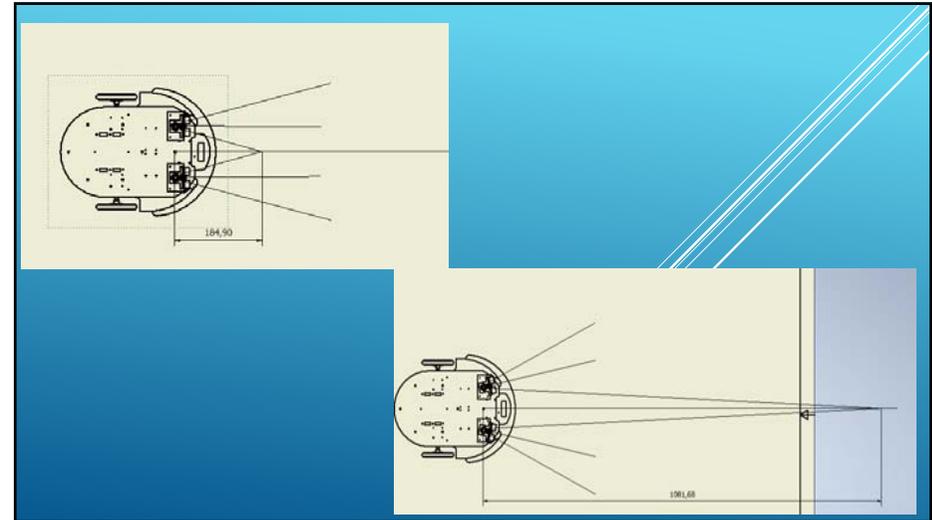
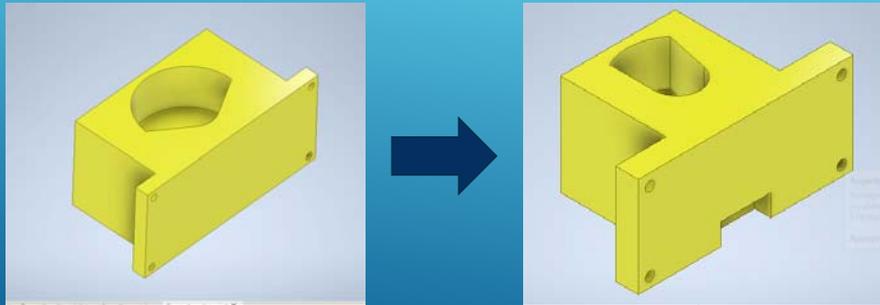
Vue de dessous



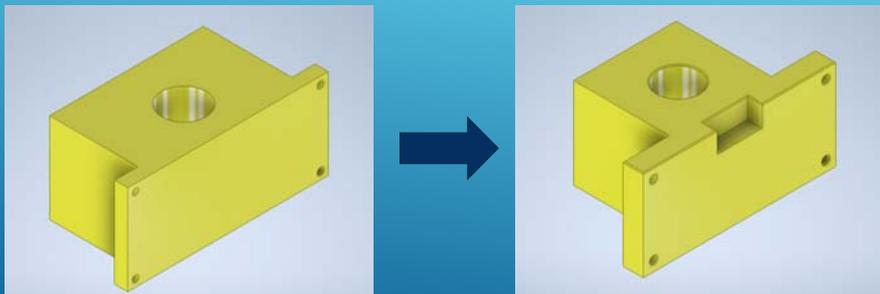
### Premier problème: la rotation du système



## DEUXIÈME PROBLÈME: ANGLE DE DÉTECTION



## TROISIÈME PROBLÈME: L'ESPACE



## RÉSULTAT:



## MON ALGORITHMME:

### Initialisation:

- ▶ si robot=vrai  
alors sounder=vrai pendant 1sec

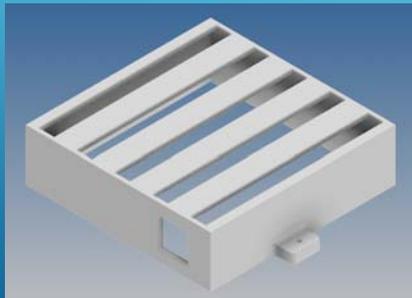
### Boucle:

- ▶ Si distG et distD > 30 et si captG et captD= faux  
alors la variable obstacle =0 et sounder=vrai pendant 1ses
- ▶ Si distG ≤ 30  
alors la variable obstacle=1 et sounder=vrai pendant 1ses
- ▶ si distD ≤ 30  
alors la variable obstacle =2 et sounder=vrai pendant 1ses
- ▶ Si distG est distD ≤ 30  
alors la variable obstacle=3 et sounder=vrai pendant 1ses

- Si catG=vrai  
alors la variable obstacle=4 et sounder=vrai pendant 1sec
- Si captD=vrai  
alors la variable obstacle =5 et sounder=vrai pendant 1ses

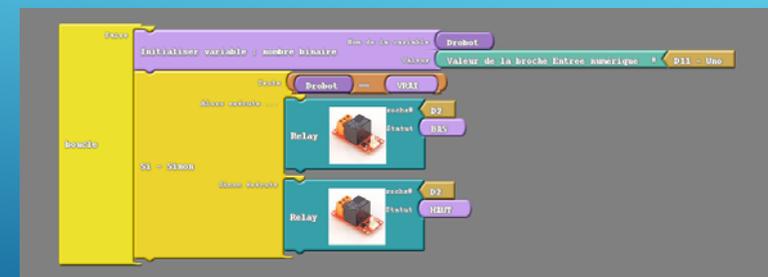
## PARTIE : SYSTÈME D'ASPIRATION DU ROBOT ASPIRATEUR

## PARTIE CONCEPTION :



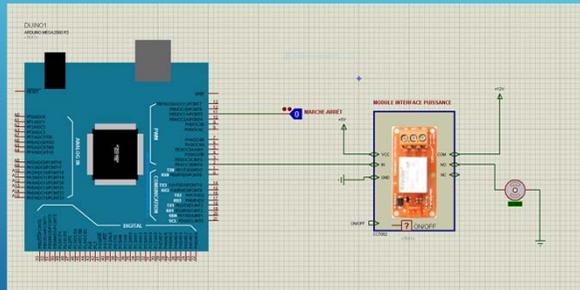
**AUTODESK**  
**INVENTOR**

## PARTIE PROGRAMMATION :

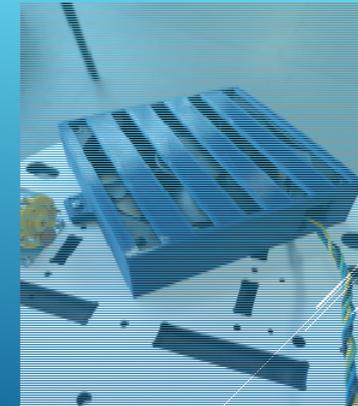


**ARDUINO**

## PARTIE SIMULATION :



## PARTIE ASSEMBLAGE :



## PARTIE : BOUTON POUSSOIR & LED



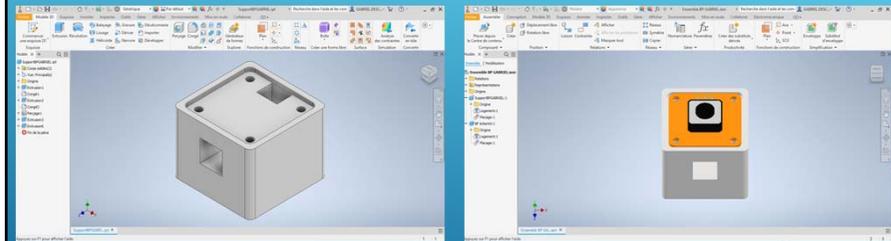
## SOMMAIRE

- 1) Conception des pièces sur Autodesk Inventor
- 2) Création du programme de déplacement sur Arduino
- 3) Simulation du programme sur ISIS 7
- 4) Montage des ensembles bouton poussoir et led

## Conception des pièces sur Autodesk Inventor



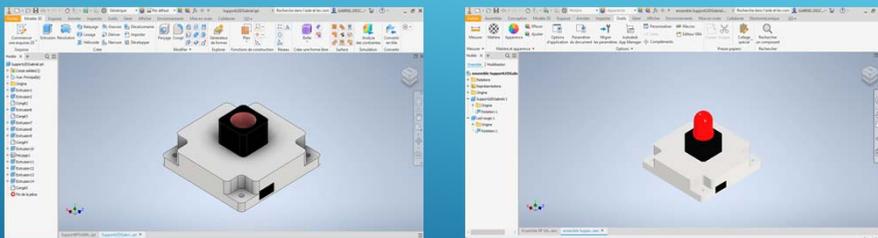
## SUPPORT BOUTON POUSSOIR



Pièce seule

Ensemble Support & bouton poussoir

## SUPPORT LED



Pièce seule

Ensemble pièce et led

## CRÉATION DU PROGRAMME DE DÉPLACEMENT SUR ARDUINO



## Création programme Arduino

This screenshot shows the initial setup of an Arduino program. It features several 'pinMode' blocks for pins 10, 11, 12, 13, 14, and 15, configuring them as either OUTPUT or INPUT. Below these, there are 'initialiser variable' blocks for binary values, with values ranging from 0 to 5.

## Création programme Arduino

This screenshot shows the beginning of a loop in the program. It starts with 'initialiser variable' blocks for binary values. This is followed by a 'tant que' (while) loop containing an 'initialiser variable' block and a 'delay' block. The loop ends with a 'fin' block.

## Création programme Arduino

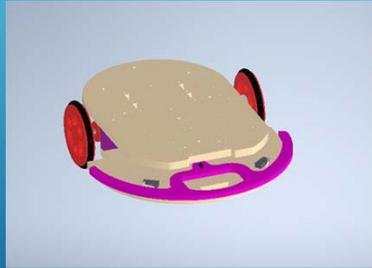
This screenshot shows the middle of the loop structure. It features 'digitalWrite' blocks for pins 10, 11, 12, 13, 14, and 15, each followed by a 'delay' block. The 'pinMode' blocks from the previous screenshot are also visible at the top of the loop.

## Création programme Arduino

This screenshot shows the end of the loop structure. It features 'digitalWrite' blocks for pins 10, 11, 12, 13, 14, and 15, each followed by a 'delay' block. The 'pinMode' blocks from the previous screenshot are also visible at the top of the loop.



### MONTAGE DES ENSEMBLES BOUTON POUSSOIR ET LED



### MONTAGE ENSEMBLES BOUTON POUSSOIR ET LED



### RÉSULTAT FINAL

