**Groupe 8 – EV3Meg Robot**



Members: LI Jiaqi, SHI Libin, TAO Kai, ZHANG Peidong

# Member and Task Introduction

## 1.1 Group Members

*SY1624127 TAO Kai (Hugo)*

*SY1624132 LI Jiaqi (Victor)*

*SY1624129 ZHANG Peidong (Paul)*

*ZY1624134 SHI Libin (Olivier)*

## 1.2 Distribution of tasks

Tracker assembly with LEGO: *all the members;*

Test of movement: *TAO Kai, ZHANG Peidong;*

Simulation with MATLAB: *all the members;*

Design of trace control and Programming: *Li Jiaqi, SHI Libin;*

Test of control: *all the members*.

# Planning

Model Selection & Robot Assembly: **17th, April**

Movement Test: **18th ~ 19th, April**

Simulation with MATLAB: **25th ~ 26th, April**

Algorithm Design & Programming: **26th, April ~ 8th, May**

Control Test: **8th ~ 11th, May**

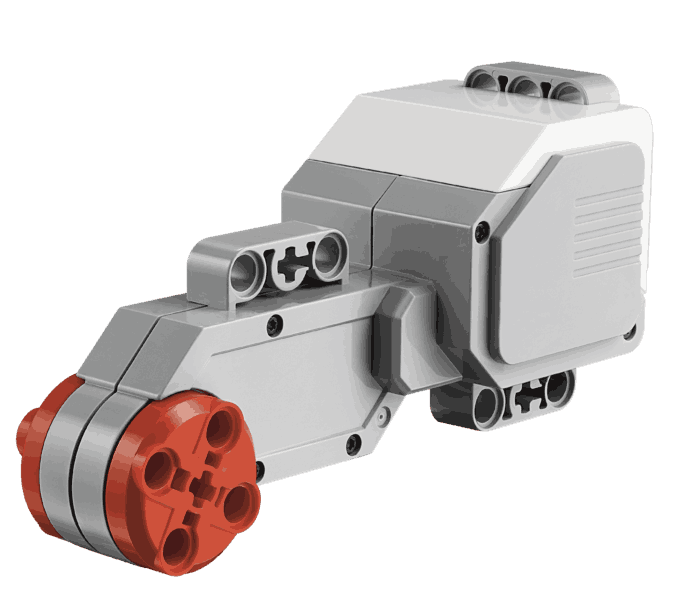
# Robot Structure

## 3.1 Components

Programmable controller x 1



large-size motors (control of wheels) x 2



medium-size motor (control of arms) x 1



ultrasonic sensor x 1



color sensor x 1



Wheels x 4



## 3.2 Assembled Robot





## 3.3 Functions

We build an automatic control tracker that can use a color sensor to follow different colors, detect objects in its way, and react.

**Function 1:** basic movements. The robot can move towards a direction, change its direction, and stop movement.

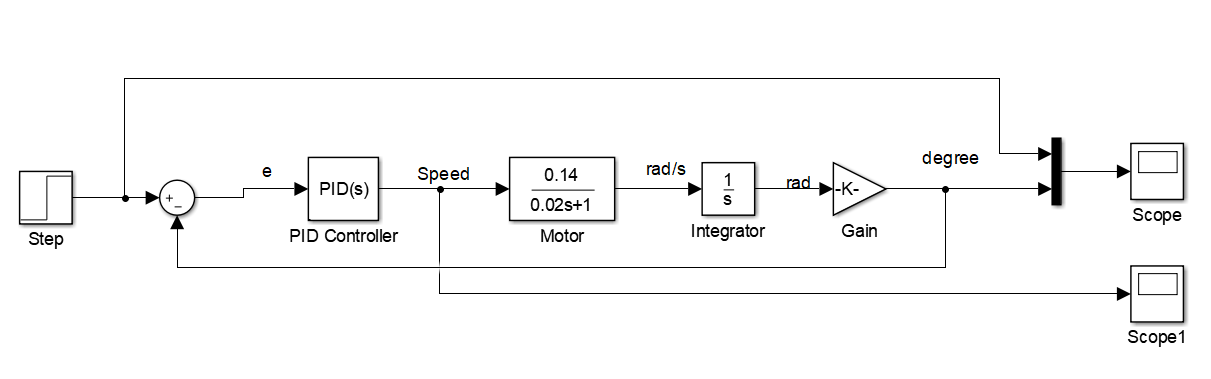
**Function 2:** use a ultrasonic sensor to avoid obstacles while moving. The robot will turn around and change its direction of movement when it detects an obstacle in its way within a predefined distance.

**Function 3:** use a color sensor to track a path of specified color. The robot will detect the color of the path and keep following the path without too much deviation.

# Evaluation of Performances

In order to realize the functions of our robot, the most basic and important one is TURN, which means control the motor to rotate a certain degree. The others because for all the functions of basic movement, obstacle avoidance and path tracking, the robot needs to turn.

Firstly, we use the simulink kit of Matlab to do the simulation, as the below figure shows:



We give the whole simulation a step function as input, representing the degree we want the motor to rotates. In this case, we need the simulation model of motor and according of the parameters given, the motor can be simulated as an one-order system and the parameters are given by our teacher. So we can use the PID to control the motor speed by mesuring error between the input and output.

During the simulation, we adjust the parameters of PID controller() to satisfy the system performance (precision, rapidity). The comparison of input and output is as the below figure shows:

# Problem and Analysis

Although we have realized the predefined three functions to a certain extent, there are still several problems that we want to discuss here. These problems occurred during the process of simulation, programming and test. Some have been solved successfully, others may yet still be improved in the future. Below are presented several problems that we think are important along with our analysis.

## 5.1 Deviation of movement along a straight line

This is probably caused by the slight inconsistence of speeds between the left wheel and the right wheel. However, this phenomenem does not appear obvious during the test process of obstacle avoiding and path tracking, and is only observable when our robot moves straight for a quite long distance (let’s say more than about 2 meters).

## 5.2 Deviation of rotation while changing direction

This problem occurs in the test process of obstacle avoiding. At first trying, the deviation of turning angle can be like 10 degrees when a comande of 90 degrees is given. After perceiving this problem, we have improved our algorithm to minimize this deviation. At the end of our test process, this deviation is reduced to less than 5 degrees given a commande of 90 degrees.

## 5.3 Lack of robustness control

A simulation using PID control is conducted in our work. However, the robustness control is left out considering the complexity of simulation. Thus, our robot may not be robust if any disturbance is involved during its movement.

# Conclusion