Robotics \_ 2020-11-25 LAB

201636417 심우석

LAB 1:

Source code:

#pragma config(Sensor, S4, ultra, sensorEV3\_Ultrasonic)

#define smooth 1.3

task main()

{

int distance, pow;

while (1)

{

distance = SensorValue(ultra);

displayTextLine(0, "%d cm", distance);

speed = abs(30 - distance) \* smooth;

// If robot is too far at obstacle

// Forward the robot

// And speed is more faster proportionally

if (distance > 30)

{

motor[motorB] = speed;

motor[motorC] = speed;

}

// If robot is too close at obstacle

// Backward the robot

// And speed is more slower proportionally

else if (distance < 30)

{

motor[motorB] = -speed;

motor[motorC] = -speed;

}

// If forces a robot be actually stopped at 30cm

else if (distance = 30)

{

motor[motorB] = 0;

motor[motorC] = 0;

}

}

move(0);

}

LAB2:

Source code:

#pragma config(Sensor, S4, ultra, sensorEV3\_Ultrasonic)

task main()

{

int degree;

float angle = 60.0;

float radius = 2.8;

float axis = 12.0/2.0;

int distance = SensorValue(ultra);

degree = (int)(axis / radius \* angle);

displayTextLine(0, "%d", degree);

resetMotorEncoder(motorB);

resetMotorEncoder(motorC);

while(1){

distance = SensorValue(ultra);

displayTextLine(0, "%d", degree);

// The distance to be measured decreases over time

// This is because the distance to the object is

// reduced because it rotates by 60 degrees and moves forward

// First obstacle turn right(Using the point turn) 60 angle

if(distance == 70)

{

setMotorTarget(motorB, -degree, 10);

setMotorTarget(motorC, degree, 10);

while(getMotorRunning(motorB)) {}

wait1Msec(1500);

}

resetMotorEncoder(motorB);

resetMotorEncoder(motorC);

// And forward to second obstacle

motor[motorB] = 20;

motor[motorC] = 20;

// Second obstacle turn left(Using the point turn) 120 angle

if (distance == 50)

{

setMotorTarget(motorB, (2\*degree), 10);

setMotorTarget(motorC, -(2\*degree), 10);

while(getMotorRunning(motorB)) {}

wait1Msec(1500);

}

resetMotorEncoder(motorB);

resetMotorEncoder(motorC);

// And forward to first obstacle

// Because, We want to go to the path, between two obstacles

motor[motorB] = 20;

motor[motorC] = 20;

// Third turn right(Using the point turn) reverse-rotate 60 angle

if (distance == 30)

{

setMotorTarget(motorB, -degree, 10);

setMotorTarget(motorC, degree, 10);

while(getMotorRunning(motorB)) {}

wait1Msec(1500);

}

resetMotorEncoder(motorB);

resetMotorEncoder(motorC);

// Go through between two obstacles

motor[motorB] = 20;

motor[motorC] = 20;

wait1Msec(2000);

// Stop the robot

motor[motorB] = 0;

motor[motorC] = 0;

}

// Stop the robot

motor[motorB] = 0;

motor[motorC] = 0;

}