#include "MyAlgo.h"

float PID(int S, float Kp, float Ki, float Kd) {

float hata, eskihata, i, P, D, pid;

eskihata = 0;

i = 0;

hata = S - 992;

P = Kp \* hata;

i = Ki\*(i + hata);

D = Kd \* (hata - eskihata);

pid = P + i + D;

eskihata = hata;

return pid;

}

bool \_\_stdcall Algo1(int S, int L, int R, int Init, int& VL, int& VR)

{

float Kp, Ki, Kd;

Kp = 0.000001;

Ki = 0.000001;

Kd = 0.01;

int hiz = 25;

float pid = PID(S, Kp, Ki, Kd);

if (S<992) {

VR = hiz + pid \* 10;

VL = hiz - pid \* 10;

}

else if (S >992) {

VR = hiz + pid \* 10;

VL = hiz - pid \* 10;

}

else {

VR = hiz + pid \* 10;

VL = hiz + pid \* 10;

}

return true;

}

bool \_\_stdcall Algo2(int S, int L, int R, int Init, int& VL, int& VR)

{

VL = 60;

VR = 60;

return true;

}

bool \_\_stdcall Algo3(int S, int L, int R, int Init, int& VL, int& VR)

{

VL = 60;

VR = 60;

return true;

}

bool \_\_stdcall Algo4(int S, int L, int R, int Init, int& VL, int& VR)

{

VL = 60;

VR = 60;

return true;

}

bool \_\_stdcall Algo5(int S, int L, int R, int Init, int& VL, int& VR)

{

VL = 60;

VR = 60;

return true;

}

bool \_\_stdcall Algo6(int S, int L, int R, int Init, double& VL, double& VR)

{

VL = 60;

VR = 60;

return true;

}

bool \_\_stdcall Algo7(int S, int L, int R, int Init, int& VL, int& VR)

{

VL = 60;

VR = 60;

return true;

}