PROGRAM AI ROBOCUP ASIA PACIFIC 2022

RESCUE CHALLENGE U19

Source Code

```
////////////////////////////////////
//
//  File : ai.c
//  CoSpace Robot
//  Version 1.0.0
//  OCT 1 2021
//  Copyright (C) 2021 CoSpace Robot. All Rights Reserved
//
////////////////////////////////////
//
// ONLY C Code can be compiled.
//
////////////////////////////////////

#define CsBot_AI_H//DO NOT delete this line
#ifndef CSBOT_REAL
#include <windows.h>
#include <stdio.h>
#include <math.h>
#define DLL_EXPORT extern __declspec(dllexport)
#define false 0
#define true 1
#endif
//The robot ID : six chars unique CID.
//Find it from your CoSpace Robot label or CoSpace program download GUI.
//Don't write the below line into two lines or more lines.
char AI_MyID[6] = {'1','2','3','4','5','6'};

int Duration = 0;
int SuperDuration = 0;
int bGameEnd = false;
int CurAction = -1;
int CurGame = 0;
int SuperObj_Num = 0;
int SuperObj_X = 0;
int SuperObj_Y = 0;
int Teleport = 0;
int LoadedObjects = 0;
int US_Front = 0;
int US_Left = 0;
int US_Right = 0;
int CSLeft_R = 0;
int CSLeft_G = 0;
int CSLeft_B = 0;
int CSRight_R = 0;
int CSRight_G = 0;
int CSRight_B = 0;
int PositionX = 0;
int PositionY = 0;
```

```
int Compass = 0;
int Time = 0;
int WheelLeft = 0;
int WheelRight = 0;
int LED_1 = 0;
int MyState = 0;
int AI_TeamID = 1;   //Robot Team ID.    1:Blue Ream;    2:Red Team.
int AI_SensorNum = 12;

#define CsBot_AI_C//DO NOT delete this line

DLL_EXPORT void SetGameID(int GameID)
{
    if(CurGame != GameID) LoadedObjects = 0;
    CurGame = GameID;
    bGameEnd = 0;
}


DLL_EXPORT void SetTeamID(int TeamID)
{
    AI_TeamID = TeamID;
}

DLL_EXPORT int GetGameID()
{
    return CurGame;
}

//Only Used by CsBot Dance Platform
DLL_EXPORT int IsGameEnd()
{
    return bGameEnd;
}

#ifndef CSBOT_REAL

char info[3000];
DLL_EXPORT char* GetDebugInfo()
{
    sprintf(info,
"Duration=%d;SuperDuration=%d;bGameEnd=%d;CurAction=%d;CurGame=%d;SuperObj_Num
=%d;SuperObj_X=%d;SuperObj_Y=%d;Teleport=%d;LoadedObjects=%d;US_Front=%d;US_Le
ft=%d;US_Right=%d;CSLeft_R=%d;CSLeft_G=%d;CSLeft_B=%d;CSRight_R=%d;CSRight_G=%
d;CSRight_B=%d;PositionX=%d;PositionY=%d;Compass=%d;Time=%d;WheelLeft=%d;Wheel
Right=%d;LED_1=%d;MyState=%d;",Duration,SuperDuration,bGameEnd,CurAction,CurGa
me,SuperObj_Num,SuperObj_X,SuperObj_Y,Teleport,LoadedObjects,US_Front,US_Left,
US_Right,CSLeft_R,CSLeft_G,CSLeft_B,CSRight_R,CSRight_G,CSRight_B,PositionX,Po
sitionY,Compass,Time,WheelLeft,WheelRight,LED_1,MyState);
    return info;
}

DLL_EXPORT char* GetTeamName()
{
     return "RCAP22ID7089";
```

```
}

DLL_EXPORT int GetCurAction()
{
    return CurAction;
}

//Only Used by CsBot Rescue Platform
DLL_EXPORT int GetTeleport()
{
    return Teleport;
}

//Only Used by CsBot Rescue Platform
DLL_EXPORT void SetSuperObj(int X, int Y, int num)
{
    SuperObj_X = X;
    SuperObj_Y = Y;
    SuperObj_Num = num;
}
//Only Used by CsBot Rescue Platform
DLL_EXPORT void GetSuperObj(int *X, int *Y, int *num)
{
    *X = SuperObj_X;
    *Y = SuperObj_Y;
    *num = SuperObj_Num;
}

#endif ////CSBOT_REAL

DLL_EXPORT void SetDataAI(volatile int* packet, volatile int *AI_IN)
{

    int sum = 0;

    US_Front = AI_IN[0]; packet[0] = US_Front; sum += US_Front;
    US_Left = AI_IN[1]; packet[1] = US_Left; sum += US_Left;
    US_Right = AI_IN[2]; packet[2] = US_Right; sum += US_Right;
    CSLeft_R = AI_IN[3]; packet[3] = CSLeft_R; sum += CSLeft_R;
    CSLeft_G = AI_IN[4]; packet[4] = CSLeft_G; sum += CSLeft_G;
    CSLeft_B = AI_IN[5]; packet[5] = CSLeft_B; sum += CSLeft_B;
    CSRight_R = AI_IN[6]; packet[6] = CSRight_R; sum += CSRight_R;
    CSRight_G = AI_IN[7]; packet[7] = CSRight_G; sum += CSRight_G;
    CSRight_B = AI_IN[8]; packet[8] = CSRight_B; sum += CSRight_B;
    PositionX = AI_IN[9]; packet[9] = PositionX; sum += PositionX;
    PositionY = AI_IN[10]; packet[10] = PositionY; sum += PositionY;
    Compass = AI_IN[11]; packet[11] = Compass; sum += Compass;
    Time = AI_IN[12]; packet[12] = Time; sum += Time;
    packet[13] = sum;

}
DLL_EXPORT void GetCommand(int *AI_OUT)
{
    AI_OUT[0] = WheelLeft;
    AI_OUT[1] = WheelRight;
```

```
    AI_OUT[2] = LED_1;
}
void TurnTo(int curRot, int targetRot)
{
    int p0 = targetRot;
    int p3 = (targetRot + 3) % 360;
    int p15 = (targetRot + 15) % 360;
    int n3 = (targetRot - 3 + 360) % 360;
    int n15 = (targetRot - 15 + 360) % 360;
    int p180 = (targetRot + 180) % 360;
    int l = 0, r = 0;
    Duration = 6;
    //Within(-3,+3)deg, stop turing.
    l = n3; r = p3;
    if ((l < r && curRot > l && curRot < r) ||
    (l > r && (curRot > l || curRot < r)))
    {
        WheelLeft = 0;
        WheelRight = 0;
        Duration = 0;
        return;
    }
    //Within[3,15]deg,Turn Slowly
    l = p3; r = p15;
    if ((l < r && curRot >= l && curRot <= r) ||
        (l > r && (curRot >= l || curRot <= r)))
    {
        WheelLeft = 10;
        WheelRight = -10;
        return;
    }
    //Within[15,180]deg,Turn Faast
    l = p15; r = p180;
    if ((l < r && curRot >= l && curRot <= r) ||
       (l > r && (curRot >= l || curRot <= r)))
    {
        WheelLeft = 30;
        WheelRight = -30;
        return;
    }
    //Within[-15,-3]deg,Turn Slowly
    l = n15; r = n3;
    if ((l < r && curRot >= l && curRot <= r) ||
    (l > r && (curRot >= l || curRot <= r)))
    {
        WheelLeft = -10;
        WheelRight = 10;
        return;
    }
    //Within[-180,-15]deg,Turn Fast
    l = p180; r = n15;
    if ((l < r && curRot >= l && curRot <= r) ||
    (l > r && (curRot >= l || curRot <= r)))
    {
        WheelLeft = -30;
```

```
        WheelRight = 30;
        return;
    }
}
void Game1()
{

    if(Duration>0)
    {
        Duration--;
    }
    else if(SuperObj_Num!=0)
    {
        Duration = 0;
        CurAction =1;
    }
    else if(CSLeft_R>=250 && CSLeft_G>=155 && CSLeft_G<=160 && CSLeft_B<=1 &&
CSRight_R>=250 && CSRight_G>=155 && CSRight_G<=161 && CSRight_B<=1 &&
Time<=29900&&(LoadedObjects>=4))
    {
        Duration = 75;
        CurAction =2;
    }
    else if(CSLeft_R>=250 && CSLeft_G>=155 && CSLeft_G<=160 && CSLeft_B<=1 &&
CSRight_R>=250 && CSRight_G>=155 && CSRight_G<=161 && CSRight_B<=1 &&
Time>=30000&&(LoadedObjects>0))
    {
        Duration = 75;
        CurAction =3;
    }
    else if(CSLeft_R>=250 && CSLeft_G>=155 && CSLeft_G<=160 &&
CSLeft_B<=1&&(LoadedObjects>2))
    {
        Duration = 0;
        CurAction =4;
    }
    else if(CSRight_R>=250 && CSRight_G>=155 && CSRight_G<=161 &&
CSRight_B<=1&&(LoadedObjects>2))
    {
        Duration = 0;
        CurAction =5;
    }
    else if(CSLeft_R>=155 && CSLeft_R<=185 && CSLeft_G>=160 && CSLeft_G<=190
&& CSLeft_B>=160 && CSLeft_B<=190)
    {
        Duration = 0;
        CurAction =6;
    }
    else if(CSLeft_R>=210 && CSLeft_R<=240 && CSLeft_G<=15 && CSLeft_B>=230
||(targetDone==2))
    {
        Duration = 65;
        CurAction =7;
    }
```

```
    else if(CSRight_R>=210 && CSRight_R<=240 && CSRight_G<=15 &&
CSRight_B>=230 ||(targetDone==2))
    {
        Duration = 65;
        CurAction =8;
    }
    else if(PositionX>1 && PositionX<15 && PositionY>1 && Compass>10 &&
Compass<80)
    {
        Duration = 0;
        CurAction =9;
    }
    else if(PositionX>=1 && PositionX<=15 && PositionY>=1 && Compass>=81 &&
Compass<=99)
    {
        Duration = 7;
        CurAction =10;
    }
    else if(PositionX>1 && PositionX<15 && PositionY>1 && Compass>100 &&
Compass<=170)
    {
        Duration = 0;
        CurAction =11;
    }
    else if(PositionX>=345 && PositionX<=360 && PositionY>=1 && Compass>=280
&& Compass<=350)
    {
        Duration = 0;
        CurAction =12;
    }
    else if(PositionX>=345 && PositionX<=360 && PositionY>=1 && Compass>=261
&& Compass<=279)
    {
        Duration = 7;
        CurAction =13;
    }
    else if(PositionX>=345 && PositionX<=360 && PositionY>=1 && Compass>=190
&& Compass<=260)
    {
        Duration = 0;
        CurAction =14;
    }
    else if(PositionX>1 && PositionY>1 && PositionY<15 && Compass>=100 &&
Compass<=170)
    {
        Duration = 0;
        CurAction =15;
    }
    else if(PositionX>=1 && PositionY>=1 && PositionY<=15 && Compass>=171 &&
Compass<=189)
    {
        Duration = 7;
        CurAction =16;
    }
```

```
    else if(PositionX>=1 && PositionY>=1 && PositionY<=15 && Compass>=190 &&
Compass<=260)
    {
        Duration = 0;
        CurAction =17;
    }
    else if(PositionX>=1 && PositionY>=255 && PositionY<=270 && Compass<=80)
    {
        Duration = 0;
        CurAction =18;
    }
    else if(PositionX>=1 && PositionY>=255 && PositionY<=270 && Compass>=350
&& Compass<=359)
    {
        Duration = 7;
        CurAction =19;
    }
    else if(PositionX>=1 && PositionY>=255 && PositionY<=270 && Compass>=280
&& Compass<=349)
    {
        Duration = 0;
        CurAction =20;
    }
    else if(CSLeft_R>=240 && CSLeft_G>=240 && CSLeft_B<=15)
    {
        Duration = 19;
        CurAction =21;
    }
    else if(CSRight_R>=240 && CSRight_G>=240 && CSRight_B<=15)
    {
        Duration = 19;
        CurAction =22;
    }
    else if(CSLeft_R>=240 && CSLeft_G<=15 && CSLeft_B<=15&&(LoadedObjects<6))
    {
        Duration = 65;
        CurAction =23;
    }
    else if(CSRight_R>=240 && CSRight_G<=15 &&
CSRight_B<=15&&(LoadedObjects<6))
    {
        Duration = 65;
        CurAction =24;
    }
    else if(CSLeft_R<=15 && CSLeft_G>=240 && CSLeft_B>=240&&(LoadedObjects<6))
    {
        Duration = 65;
        CurAction =25;
    }
    else if(CSRight_R<=15 && CSRight_G>=240 &&
CSRight_B>=240&&(LoadedObjects<6))
    {
        Duration = 65;
        CurAction =26;
    }
```

```
    else if(CSRight_R>=250 && CSRight_G>=3 && CSRight_G<=4 && CSRight_B>=240
&& CSRight_B<=244)
    {
        Duration = 65;
        CurAction =27;
    }
    else if(CSLeft_R>=250 && CSLeft_G>=3 && CSLeft_G<=4 && CSLeft_B>=244 &&
CSLeft_B<=244)
    {
        Duration = 0;
        CurAction =28;
    }
    else if(CSLeft_R<=15 && CSLeft_G<=15 && CSLeft_B<=15&&(LoadedObjects<6))
    {
        Duration = 65;
        CurAction =29;
    }
    else if(CSRight_R<=15 && CSRight_G<=15 &&
CSRight_B<=15&&(LoadedObjects<6))
    {
        Duration = 65;
        CurAction =30;
    }
    else if(PositionX<=288 && PositionY<=249&&(gotoTarget==1 ||
gotoTarget==2))
    {
        Duration = 0;
        CurAction =31;
    }
    else if(targetDone==3)
    {
        Duration = 19;
        CurAction =32;
    }
    else if(gotoTarget==0)
    {
        Duration = 0;
        CurAction =33;
    }
    else if(gotoTarget>0)
    {
        Duration = 0;
        CurAction =34;
    }
    switch(CurAction)
    {
        case 1:
            WheelLeft=0;
            WheelRight=0;
            LED_1=0;
            S_X=SuperObj_X;

            S_Y=SuperObj_Y;

gotoTarget=2;
```

```
        break;
    case 2:
        WheelLeft=0;
        WheelRight=0;
        LED_1=2;
        if(Duration == 1) {LoadedObjects = 0;}
        if(Duration < 16)
        {
            WheelLeft = -50;
            WheelRight = -50;
        }
        if(Duration < 8)
        {
            WheelLeft = -50;
            WheelRight = 50;
        }
        gotoTarget=0;

        break;
    case 3:
        WheelLeft=0;
        WheelRight=0;
        LED_1=2;
        if(Duration == 1) {LoadedObjects = 0;}
        if(Duration < 16)
        {
            WheelLeft = -50;
            WheelRight = -50;
        }
        if(Duration < 8)
        {
            WheelLeft = -50;
            WheelRight = +50;
        }
        gotoTarget=0;

        break;
    case 4:
        WheelLeft=10;
        WheelRight=40;
        LED_1=0;
        gotoTarget=0;

        break;
    case 5:
        WheelLeft=40;
        WheelRight=10;
        LED_1=0;
        gotoTarget=0;

        break;
    case 6:
        WheelLeft=100;
        WheelRight=100;
```

```
            LED_1=0;
            break;
        case 7:
            WheelLeft=0;
            WheelRight=0;
            LED_1=1;
            if(Duration == 1) LoadedObjects++;
            if(Duration < 6)
            {
                WheelLeft = 40;
                WheelRight = 40;
            }
            gotoTarget=0;

targetDone=0;

            break;
        case 8:
            WheelLeft=0;
            WheelRight=0;
            LED_1=1;
            if(Duration == 1) LoadedObjects++;
            if(Duration < 6)
            {
                WheelLeft = 40;
                WheelRight = 40;
            }
            gotoTarget=0;

targetDone=0;

            break;
        case 9:
            WheelLeft=0;
            WheelRight=-70;
            LED_1=2;
            break;
        case 10:
            WheelLeft=0;
            WheelRight=0;
            LED_1=2;
            if(Duration>4)
{
 WheelLeft=-80;
                    WheelRight=-80;

}
else
{
 if(Compass>90)
 {
   WheelLeft=-80;
                    WheelRight=80;

 }
```

```
 else
 {
   WheelLeft=80;
                     WheelRight=-80;

 }
}
            break;
        case 11:
            WheelLeft=-70;
            WheelRight=0;
            LED_1=2;
            break;
        case 12:
            WheelLeft=-70;
            WheelRight=0;
            LED_1=2;
            break;
        case 13:
            WheelLeft=0;
            WheelRight=0;
            LED_1=2;
            if(Duration>4)
{
 WheelLeft=-80;
                     WheelRight=-80;

}
else
{
 if(Compass>270)
 {
   WheelLeft=-80;
                     WheelRight=80;

 }
 else
 {
   WheelLeft=80;
                     WheelRight=-80;

 }
}
            break;
        case 14:
            WheelLeft=0;
            WheelRight=-70;
            LED_1=2;
            break;
        case 15:
            WheelLeft=0;
            WheelRight=-70;
            LED_1=0;
            break;
        case 16:
```

```
            WheelLeft=0;
            WheelRight=0;
            LED_1=2;
            if(Duration>4)
{
 WheelLeft=-80;
                    WheelRight=-80;

}
else
{
 if(Compass>180)
 {
   WheelLeft=-87;
                    WheelRight=87;

 }
 else
 {
   WheelLeft=87;
                    WheelRight=-87;

 }
}
            break;
        case 17:
            WheelLeft=-70;
            WheelRight=0;
            LED_1=2;
            break;
        case 18:
            WheelLeft=-70;
            WheelRight=0;
            LED_1=2;
            break;
        case 19:
            WheelLeft=0;
            WheelRight=0;
            LED_1=2;
            if(Duration>4)
{
 WheelLeft=-87;
                    WheelRight=-87;

}
else
{
 if(Compass>0)
 {
   WheelLeft=-87;
                    WheelRight=87;

 }
 else
 {
```

```
   WheelLeft=87;
                     WheelRight=-87;

 }
}
            break;
        case 20:
            WheelLeft=0;
            WheelRight=-70;
            LED_1=2;
            break;
        case 21:
            WheelLeft=20;
            WheelRight=-40;
            LED_1=0;
            if(Duration<10)
{
 WheelLeft=60;
                     WheelRight=60;

}
            break;
        case 22:
            WheelLeft=-40;
            WheelRight=20;
            LED_1=0;
            if(Duration<10)
{
 WheelLeft=60;
                     WheelRight=60;

}

            break;
        case 23:
            WheelLeft=0;
            WheelRight=0;
            LED_1=1;
            if(Duration == 1) LoadedObjects++;
            if(Duration < 6)
            {
                WheelLeft = 40;
                WheelRight = 40;
            }
            break;
        case 24:
            WheelLeft=0;
            WheelRight=0;
            LED_1=1;
            if(Duration == 1) LoadedObjects++;
            if(Duration < 6)
            {
                WheelLeft = 40;
                WheelRight = 40;
            }
```

```
            break;
        case 25:
            WheelLeft=0;
            WheelRight=0;
            LED_1=1;
            if(Duration == 1) LoadedObjects++;
            if(Duration < 6)
            {
                WheelLeft = 40;
                WheelRight = 40;
            }
            break;
        case 26:
            WheelLeft=0;
            WheelRight=0;
            LED_1=1;
            if(Duration == 1) LoadedObjects++;
            if(Duration < 6)
            {
                WheelLeft = 40;
                WheelRight = 40;
            }
            break;
        case 27:
            WheelLeft=0;
            WheelRight=0;
            LED_1=1;
            if(Duration == 1) LoadedObjects++;
            if(Duration < 6)
            {
                WheelLeft = 40;
                WheelRight = 40;
            }
            break;
        case 28:
            WheelLeft=0;
            WheelRight=0;
            LED_1=0;
            break;
        case 29:
            WheelLeft=0;
            WheelRight=0;
            LED_1=1;
            if(Duration == 1) LoadedObjects++;
            if(Duration < 6)
            {
                WheelLeft = 70;
                WheelRight = 70;
            }
            break;
        case 30:
            WheelLeft=0;
            WheelRight=0;
            LED_1=1;
            if(Duration == 1) LoadedObjects++;
```

```
            if(Duration < 6)
            {
                WheelLeft = 70;
                WheelRight = 70;
            }
            break;
        case 31:
            WheelLeft=0;
            WheelRight=0;
            LED_1=0;
            gotoTarget=3;

            break;
        case 32:
            WheelLeft=0;
            WheelRight=0;
            LED_1=0;
            gotoTarget=1;
                     targetDone=0;

            break;
        case 33:
            WheelLeft=0;
            WheelRight=0;
            LED_1=0;
            if (US_Front < 12)
{
  if (US_Left > US_Right)
  {
    WheelLeft = -50;
                     WheelRight = 50;

  }
  else
  {
    WheelLeft = 50;
                     WheelRight = -50;

  }
}
else if (US_Left < 15)
{
  WheelLeft = 50;
                     WheelRight = -30;

}
else if (US_Right < 15)
{
  WheelLeft = -30;
                     WheelRight = 50;

}
else if (US_Front < 20)
{
  WheelLeft = 70;
```

```
                    WheelRight = 70;

}
else
{
  WheelLeft = 100;
                    WheelRight = 100;

}

if(LoadedObjects>=5)
{
  gotoTarget=1;
                    //deposit
}

if((Time>30000) && (LoadedObjects>=2))
{
  gotoTarget=1;

}
            break;
        case 34:
            WheelLeft=0;
            WheelRight=0;
            LED_1=0;
            #define RADS 57.2958

int xPos = PositionX;

int yPos = PositionY;

int Kp = 2, Kd = 2, speed = 100, PIDHeading = 0;

float x, y, atanBuff, targetDistance, targetDistance1, targetDistance2,
targetBearing, headingError, dHeadingError, theta;

int xDeposit1, xDeposit2, yDeposit1, yDeposit2;

int xSafeLoc1, ySafeLoc1,xSafeLoc2, ySafeLoc2,xSafeLoc3, ySafeLoc3,xSafeLoc4,
ySafeLoc4;


xSafeLoc1=194;
                    ySafeLoc1=182;

xSafeLoc2=219;
                    ySafeLoc2=244;

xSafeLoc3=0;
                    ySafeLoc3=0;

xSafeLoc4=0;
                    ySafeLoc4=0;
```

```
if(gotoTarget==1)//deposit
{
  xDeposit1=346;
                    yDeposit1=187;

  xDeposit2=165;
                    yDeposit2=76;


  xTarget = xDeposit1;
                    yTarget = yDeposit1;
                    //deposit atas
  x = xTarget - xPos;

  y = yTarget - yPos;

  targetDistance1 = sqrt((x*x) + (y*y));


  xTarget = xDeposit2;
                    yTarget = yDeposit2;
                    //deposit bawah
  x = xTarget - xPos;

  y = yTarget - yPos;

  targetDistance2 = sqrt((x*x) + (y*y));


  if(targetDistance1<targetDistance2)
  {
      xTarget = xDeposit1;
                    yTarget = yDeposit1;
                    //atas
  }
  else
  {
      xTarget = xDeposit2;
                    yTarget = yDeposit2;
                    //bawah
  }
}
else if(gotoTarget==2)//superObject
{
  xTarget = S_X;

  yTarget = S_Y;

}
else if(gotoTarget==3)//lokasi aman1
{
  xTarget = xSafeLoc1;

  yTarget = ySafeLoc1;
```

```
}
else if(gotoTarget==4)//lokasi aman2
{
  xTarget = xSafeLoc2;

  yTarget = ySafeLoc2;

}
else if(gotoTarget==5)//lokasi aman3
{
  xTarget = xSafeLoc3;

  yTarget = ySafeLoc3;

}
else if(gotoTarget==6)//lokasi aman4
{
  xTarget = xSafeLoc4;

  yTarget = ySafeLoc4;

}

x = xTarget - xPos;

y = yTarget - yPos;

atanBuff = atan(y / x) * RADS;


targetDistance = sqrt((x * x) + (y * y));

if (x > 0.00001) targetBearing = 90.0 - atanBuff;

else if (x < -0.00001) targetBearing = -90.0 - atanBuff;


if (Compass > 180.0)theta = 360.0 - Compass;

else if (Compass < 180.0)theta = - Compass;


headingError = targetBearing - theta;

if (headingError > 180.0) headingError -= 360.0;

else if (headingError < -180.0) headingError += 360.0;



dHeadingError = headingError - lastHeadingError;

lastHeadingError = headingError;
```

```
PIDHeading = Kp * headingError + Kd * dHeadingError;

if (PIDHeading > 70)PIDHeading = 70;

else if (PIDHeading < -70)PIDHeading = -70;


if (abs(targetDistance) < 12)
{
  WheelLeft = 0;
                    WheelRight = 0;

  if(gotoTarget==1)targetDone=1;

  else if(gotoTarget==2)targetDone=2;

  else if(gotoTarget==3)targetDone=3;

  else if(gotoTarget==4)targetDone=4;

  else if(gotoTarget==5)targetDone=5;

  else if(gotoTarget==6)targetDone=6;

}
else
{
  if (US_Front < 12)
  {
    if (US_Left > US_Right)
    {
      WheelLeft = -40;
                    WheelRight = 50;

    }
    else
    {
      WheelLeft = 50;
                    WheelRight = -40;

    }
  }
  else if (US_Left < 15)
  {
    WheelLeft = 50;
                    WheelRight = -20;

  }
  else if (US_Right < 15)
  {
    WheelLeft = -20;
                    WheelRight = 50;

  }
  else
```

```
  {
    if (abs(headingError) > 50)speed = 0;

    if(xPos==0 && yPos==0) //area signal block zone
    {
        WheelLeft = 80;

        WheelRight = 80;

    }
    else
    {
        WheelLeft = speed + PIDHeading;

        WheelRight = speed - PIDHeading;

    }
  }
}

targetDistanceBuff = targetDistance;

headingErrorBuff = headingError;

targetBearingBuff = targetBearing;

thetaBuff = theta;

            break;
        default:
            break;
    }

}


DLL_EXPORT void OnTimer()
{
    switch (CurGame)
    {
        case 9:
            break;
        case 10:
            WheelLeft=0;
            WheelRight=0;
            LED_1=0;
            break;
        case 0:
            Game0();
            break;
        case 1:
            Game1();
            break;
        default:
            break;
```

```
    }
}

//juara 1
```