עומר פלדמן 302883970

אור שלו 305758377

דו"ח פרויקט DCS

1. אלגוריתם עקיפת המכשולים –

נסיעה ישר באמצעות הפונקציה Go() תוך כדי בדיקה קבועה של חיישני הIR

לא

בדיקת האם אחד החיישנים מזהה מתח הגבוה מ2.3V (האם יש מכשול מקדימה)

כן

בדיקת מיקום הרכב על ציר הX ועצירת הרכב עם פונקציה Stop()

הרכב משמאל למרכז ציר הX

הרכב מימין למרכז ציר הX

פניה ימינה וסיבוב סרבו שמאל ב90 מעלות באמצעות הפונקציה TurnRight() והתחלת התקדמות באמצעות Go()

פניה שמאלה וסיבוב סרבו ימין ב90 מעלות באמצעות הפונקציה TurnLeft() והתחלת התקדמות באמצעות Go()

גדול מV1.8

גדול מV1.8

כניסה ללולאה הבודקת אם הערך בIR שמאל גדול מ1.8V (האם עדיין ממול למכשול)

כניסה ללולאה הבודקת אם הערך בIR ימין גדול מ1.8V (האם עדיין ממול למכשול)

קטן מV1.8

קטן מV1.8

יציאה מהלולאה, המשך נסיעה קצרה על מנת להעביר את כל גוף הרכב, פנייה ימינה לכיוון היציאה וסיבוב סרבו ימין חזרה לקדמה באמצעות פונקציית FixRight()

יציאה מהלולאה, המשך נסיעה קצרה על מנת להעביר את כל גוף הרכב, פנייה שמאלה לכיוון היציאה וסיבוב סרבו שמאל חזרה לקדמה באמצעות פונקציית FixLeft()

המשך נסיעה ישר באמצעות הפונקציה Go() תוך כדי בדיקה קבועה של חיישני הIR

הקוד לעקיפת מכשולים (ערך מרכז ציר X הוא 5060)

**if**((ADCMeas[0]>=2.3 || ADCMeas[1]>=2.3) && x>=5060){

Stop();

waitt(5);

TurnLeft();

waitt(6);

Go()

**while**(ADCMeas[1]>=1.8);

waitt(14);

Stop();

waitt(5);

FixRight();

waitt(12);

}

**if**((ADCMeas[0]>=2.3 || ADCMeas[1]>=2.3) && x < 5060){

Stop();

waitt(5);

TurnRight();

waitt(6);

Go();

**while**(ADCMeas[0]>=1.8);

waitt(14);

Stop();

waitt(5);

FixLeft();

waitt(40);

}

**void** **Go**()

{

TPM2\_C1V = 0x3000;

TPM1\_C0V = 0x3880;

GPIOC\_PCOR |= 0XFFF;

GPIOC\_PSOR |= 0X440;

}

**void** **TurnRight**()

{

turnR=1;

TurnCount=0;

TurnAngle = 87;

dir = (dir+1)%4;

GPIOC\_PCOR |= 0XFFF;

GPIOC\_PSOR |= 0X420;

TPM1\_C1V = 0x870;

**while** (turnR ==1) k++;

}

**void** **TurnLeft**()

{

turnL=1;

TurnCount=0;

TurnAngle = 87;

dir = (dir+3)%4;

GPIOC\_PCOR |= 0XFFF;

GPIOC\_PSOR |= 0XC0;

TPM0\_C4V = 0x740;

}

**void** **FixRight**()

{

turnR=1;

TurnCount=0;

TurnAngle = 87;

dir = (dir+1)%4;

GPIOC\_PCOR |= 0XFFF;

GPIOC\_PSOR |= 0X420;

TPM0\_C4V = 0xC95;

}

**void** **FixLeft**()

{

turnL=1;

TurnCount=0;

TurnAngle = 87;

dir = (dir+3)%4;

GPIOC\_PCOR |= 0XFFF;

GPIOC\_PSOR |= 0XC0;

TPM1\_C1V = 0x385;

}

1. תהליך קבלת החלטות כפונקציה של דגימת החיישנים-

הרכב מקבל החלטות כפונקציה של דגימת החיישנים בשני מקרים.

הראשון – בעת עקיפת מכשולים. תהליך זה מפורט בסעיף א'.

השני – עבור יציאה מהפתחים בחצי העיגול. את מקרה זה אנו נפרט בסעיף זה.

סריקת המעגל בעזרת 2 החיישנים ושמירת ערכי הקליטה במערך

המרת הערכים למרחקים וחיפוש אחר הפתח

זיהוי בין אילו זוויות הפתח נמצא וכיוון ושמרית ערך זווית מרכז הפתח

שימוש בזווית זו על מנת לפנות אל הפתח ולהתיישר לכיוון היציאה בעזרת הפונקציות TurnAngleL ו-TurnAngleR

**void** **ScanForOpen**()

{

OpeningR=0;

circle = 1;

**while**(SetRotate==0 && i<23){

**if**(TPM0\_C4V <= 0x740){

SetRotate=1;

circle = 0;

}

}

TPM1\_C1V = 0x370;

TPM0\_C4V = 0xCD5;

SetRotate =0;

**for** (i=0;i<23;i++){

Sin = sin(PI/2-i\*PI/46);

Cos = cos(PI/2-i\*PI/46);

**int** k;

**for** (k=0; k<14; k++ ){

**if**(samp[0][i] <= IRDist[k] && samp[0][i] > IRDist[k+1]){

placeL=k;

}

**if**(samp[1][i] <= IRDist[k] && samp[1][i] > IRDist[k+1]){

placeR=k;

}

}

samp[0][i] = 20+placeL\*10 + (1-((samp[0][i]-IRDist[placeL+1])/(IRDist[placeL]-IRDist[placeL+1])))\*10; //Left Opening

**if** ((samp[0][i] > (OpenL[0]+60)) && i!=0){

OpenEndL = 1;

**if**(OpeningL ==0) OpeningL = i;

}

**else** **if**(OpenEndL==1){

OpeningL+=i-1;

OpeningL = OpeningL/2;

OpenL[0] = samp[0][OpeningL];

OpenL[1] = OpeningL\*4.1;

OpenEndL=0;

}

**else** OpenL[0]=samp[0][i];

samp[1][i] = 20+placeR\*10 + (1-((samp[1][i]-IRDist[placeR+1])/(IRDist[placeR]-IRDist[placeR+1])))\*10; //Right Opening

**if** ((samp[1][i] > 120) && i!=0){ //120 = OpenR[0]+40

OpenEndR = 1; //Start of the opening

**if**(OpeningR ==0){

OpeningR = i; //degree of the start of the opening

}

}

**else** **if**(OpenEndR==1){

OpeningR+=i-2;

OpeningR = OpeningR/2;

**if**(OpeningR<2) OpeningR=0;

OpenR[0] = samp[1][OpeningR];

OpenR[1] = OpeningR\*4.1;

OpenEndR=0;

}

**else** OpenR[0]=samp[1][i];

**if** (OpenEndR==1 && i==22){ // in case we are looking at 90 degrees

OpeningR+=i-2;

OpeningR = OpeningR/2;

**if**(OpeningR<2) OpeningR=0;

OpenR[0] = samp[1][OpeningR];

OpenR[1] = OpeningR\*4.1;

OpenEndR=0;

}

placeL=-1;

placeR=-1;

}

}

**void** **TurnAngleL** (**double** angle)

{

turnL=1;

TurnCount=0;

TurnAngle = OpenR[1]-4;

GPIOC\_PCOR |= 0XFFF;

GPIOC\_PSOR |= 0XC0;

}

**void** **TurnAngleR** (**double** angle)

{

turnR=1;

TurnCount=0;

TurnAngle = OpenR[1]-2;

GPIOC\_PCOR |= 0XFFF;

GPIOC\_PSOR |= 0X420;

}