International Journal of
Automotive Manufacturing and Materials (IJAMM)

Scilight

Article

A Control System Design for an Intelligent Unmanned Automotive

Yundi Yang †, Xin Gao †, Jinwen You, Dengbo Zhang, Zhuo Zhang and Yuanmei Song *

School of Mechanical & Vehicle Engineering, Linyi University, Linyi 276000, China
* Correspondence: ymsong321@163.com
† These authors contribute equally to this work.
Received: 1 August 2024; Revised: 18 September 2024; Accepted: 28 November 2024; Published: 4 December 2024

Supplementary Materials

DDA arc interpolation program:

X_P	BIT 00H;	X-direction overflow flag
Y_P	BIT 01H;	Y-direction overflow flag
X_S	EQU 60H;	Starting point coordinate X
Y_S	EQU 61H;	Starting point coordinate Y
X_E	EQU 62H;	Endpoint coordinate X
Y_E	EQU 63H;	Endpoint coordinate Y
J_{VX}	EQU 64H;	X Integral Accumulator
J_{VY}	EQU 65H ;	Y Integral Accumulator
J_{RX}	EQU 66H;	X integrand function register
J_{RY}	EQU 67H;	Y integrand function register
J_{EX}	EQU 68H;	X-direction endpoint counter
J_{EY}	EQU 69H;	Y-direction endpoint counter
ORG	1000H	
MOV	J_{VX}, YS;	initialization
MOV	J_{VY}, XS	
MOV	J_{RX}, #0	
MOV	J_{RY}, #0	
MOV	R2, X_S	
MOV	R4, X_E	
ACALL	BSUB;	Find the initial count value for the X coordinate
MOV	J_{EX}, R6	
MOV	R2, Y_S	
MOV	R4, Y_E	
ACALL	BSUB;	Find the initial count value for the Y coordinate
MOV	J_{EY}, R6	
CLR	X_P	
CLR	Y_P	
MOV	R2, X_S	
MOV	R4, Y_S	

```
    ACALL    YC;                 Call overflow subroutine
CF: MOV      A, JEX;             X direction
    JZ       YX
    MOV      R2, JRX
    MOV      R4, JVX
    ACALL    BADD;               Modify the X-direction register
    MOV      JRX, R6
    MOV      A, R7
CJNE         A, JRX, NX1;        Does the X-direction overflow
    SETB     XP
    DEC      XS
    DEC      JEX;                -X takes a step forward
    AJMP     YX
    NX1: JC  YX
      SETB   XP
      DEC    XS
    DEC      JEX
XY: MOV      A, JEY;             Y direction
JZ           ZDP
    MOV      R2, JRY
    MOV      R4, JVY
    ACALL    BADD;               Modify the Y-direction register
    MOV      JRY, R6
    MOV      A, R7
      CJNE   A, JRY, NX2         Does the Y-direction overflow
    SETB     YP;                 +Y takes a step forward
    INC      YS
    DEC      JEY
    AJMP     JINX
NX2: JC      JINX;               Feed X?
    SETB     YP
    INC      YS
    DEC      JEY
JINX: JNB    XP, NX3;            Feed Y?
    DEC      JVY
NX3: JNB     YP, CF
    INC      JVX
    AJMP     CF
ZDP: MOV     A, JEX
    JNZ      CF;                 Is X reaching the finish line?
    MOV      A, JEY
    JNZ      CF;                 Is Y reaching the finish line?
    END
```

```
BADD: Addition program entrance; Sum added R2; Add R4; Result R6;
BSUB: Subtraction program entrance; Meiotic R2; Reduced R4; Result R6.
BSUB: MOV     A, R4;          Take meiosis
    CPL       ACC.7;          Reverse the meiotic symbol for addition
    MOV       R4, A
BADD: MOV     A, R2;          Take the added number
    XRL       A, R4;          Two numbers with different numbers
MOV C, ACC.7; CY=0 for two identical numbers, CY=1 for two identical numbers
    MOV       A, R2
    CLR       ACC.7;          Symbol clarity 0
    MOV       R2, A
    MOV       A, R4
    CLR       ACC.7;          Symbol clarity 0
    MOV       R4, A
    JC        JIAN;           Convert two numbers with different numbers to JIAN MOV
              A, R2
    ADD       A, R4
    MOV       R6, A
    RET
JIAN: MOV     A, R2;          subtract
    CLR       C
    SUBB      A, R4
    MOV       R6, A
    JNB       ACC.7, QWE
    MOV       A, R6
    CPL       A
    ADD       A, #1
    MOV       R6, A
QWE: RET
    Overflow subroutine: R7 stores overflow values.
    YC: MOV   R5, #08H
      MOV     R7, #00H
      CLR     C
      MOV     A, R2
      SUBB    A, R4
      JNZ     LP
    LP5: MOV  A, R2
    LP6: CLR  C
    LP2: RLC  A
      INC     R7
      DJNZ    R5, LP2
LP1: CLR      A
LP3: SETB     C
```

```
    RRC     A
    DJNZ    R7, LP3
    CPL     A
    MOV     R7, A
    RET
LP: JC      LP4
    AJMP    LP5
LP4: MOV    A, R4
    AJMP    LP6
```