

บทที่ 3

วิธีดำเนินการวิจัยคิดค้นสิ่งประดิษฐ์

โครงการวิจัยคิดค้นประดิษฐ์สิ่งใหม่บทนี้ได้กล่าวถึง ขั้นตอนการดำเนินงาน รูปลักษณะ
ต้นแบบเครื่องบรรจุไข่อัตโนมัติ Ladder Diagram ควบคุมเครื่องบรรจุไข่อัตโนมัติ และ
โปรแกรมควบคุมการทำงานของเครื่องบรรจุไข่ให้เป็นอัตโนมัติ ตามลำดับ

ขั้นตอนการดำเนินงาน

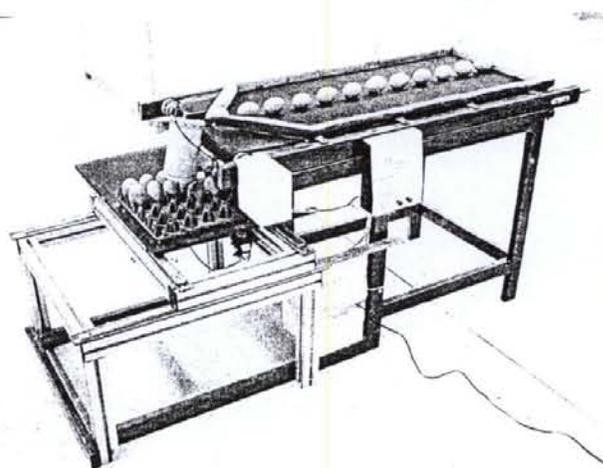
โครงการนี้มีระยะเวลาดำเนินการ 12 เดือน เริ่มตั้งแต่เดือน ตุลาคม พ.ศ. 2547 จนถึง
กันยายน พ.ศ. 2548 ซึ่งมีขั้นตอนการดำเนินงานดังนี้

1. คณะผู้วิจัยร่วมมือกับผู้เชี่ยวชาญระดมแนวความคิดเพื่อกำหนดเงื่อนไขในการออกแบบ
2. ออกแบบและเขียนแบบของต้นแบบพร้อมทำการคำนวณเบื้องต้น
3. จำลองการทำงาน ด้วยโปรแกรมคอมพิวเตอร์กราฟิกประยุกต์ทางวิศวกรรม
4. เขียนแบบพร้อมสร้าง
5. วางข้อกำหนดของแบบพร้อมสร้าง แล้วติดต่อหาผู้แทนจำหน่ายต่างๆเพื่อประเมินราคา

และเวลานำส่งของต่างๆ

6. สร้างต้นแบบ
7. ทดสอบการทำงาน และเก็บรวบรวมข้อมูล
8. ทำการปรับปรุงแก้ไขให้ได้การทำงานที่พึงประสงค์และมีประสิทธิภาพสูงสุด
9. สรุปและจัดทำรายงานการวิจัย

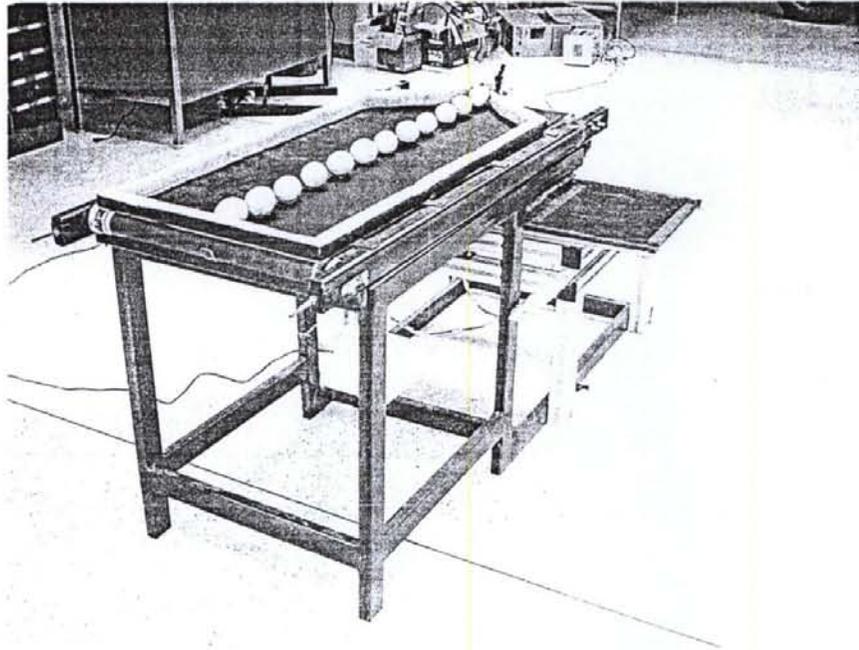
รูปลักษณะต้นแบบของเครื่องบรรจุไข่อัตโนมัติ



รูปที่ 3.1 แสดงรูปลักษณะต้นแบบของเครื่องบรรจุไข่อัตโนมัติ

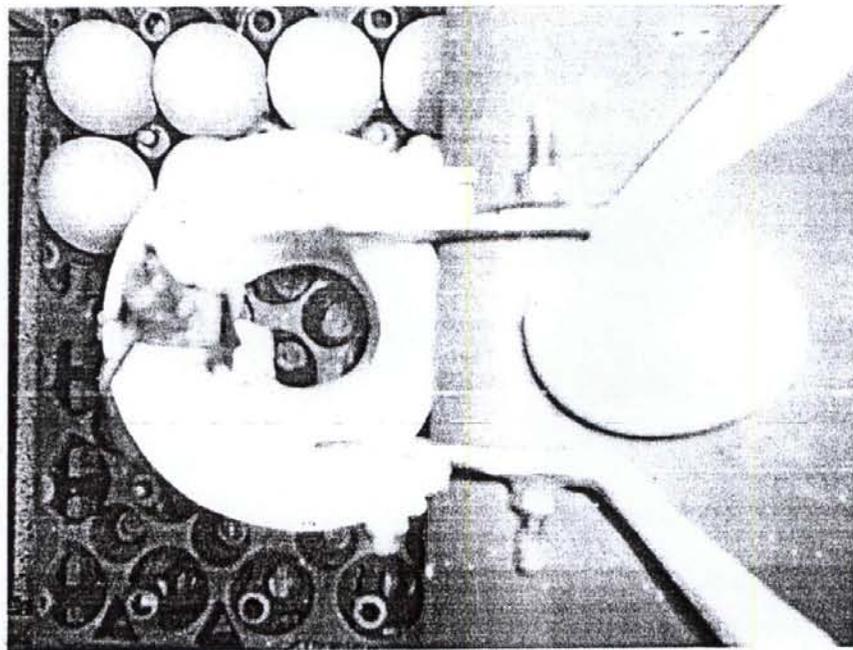
ต้นแบบเครื่องบรรจุไข่อัตโนมัตินี้ประกอบด้วย 4 ส่วนหลักคือ

1. สายพานลำเลียงไข่ ทำหน้าที่ลำเลียงไข่เข้าปล่องบรรจุไข่ลงถาด



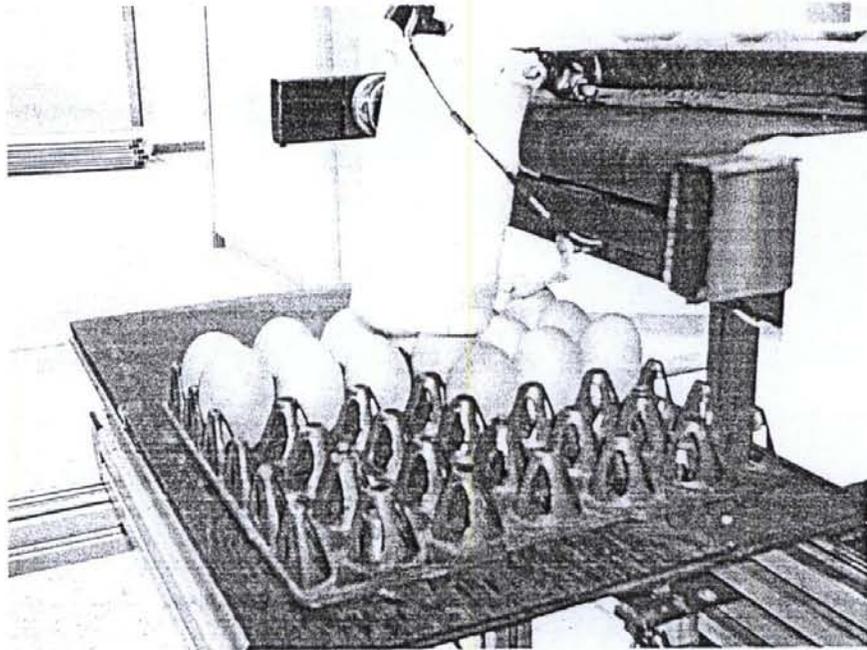
รูปที่ 3.2 แสดงรูปสายพานลำเลียงไข่

2. ปล่องบรรจุไข่ลงถาด ทำหน้าที่นำและส่งไข่ให้ลงไปอยู่ในถาด



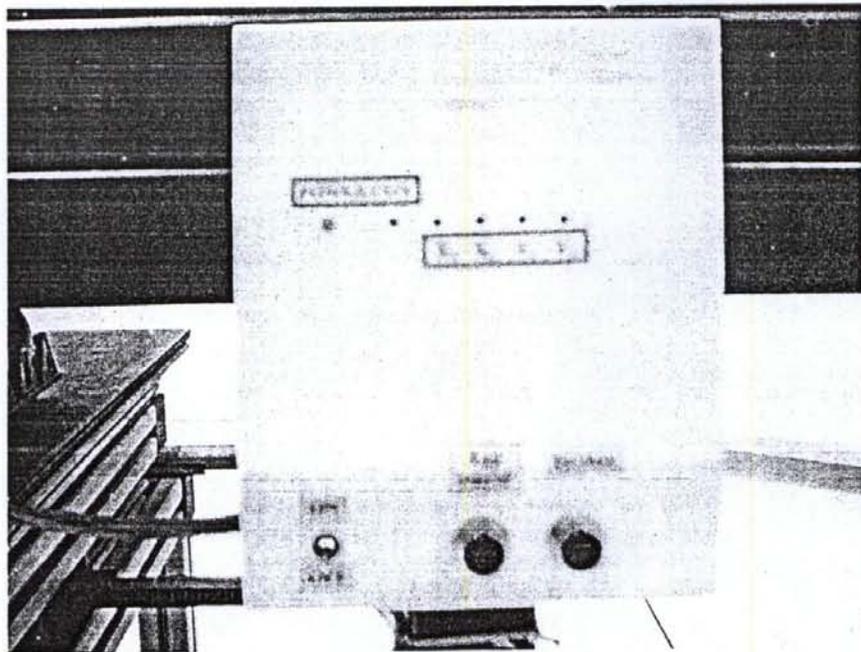
รูปที่ 3.3 แสดงรูปปล่องบรรจุไข่ลงถาด

3. โต๊ะ x-y ทำหน้าที่เลื่อนถาดวางไข่ให้เข้าไปรับไข่จากปล่อง



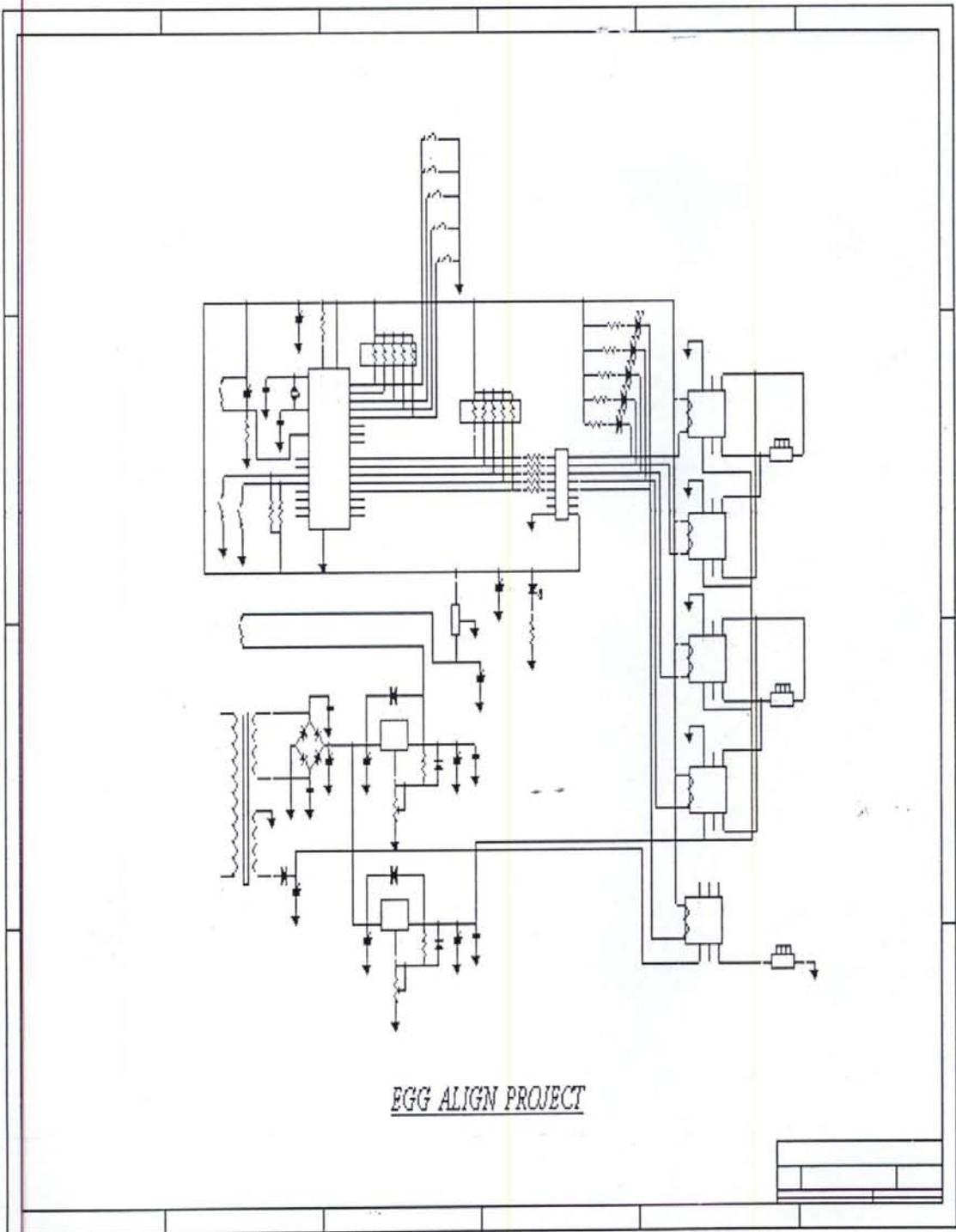
รูปที่ 3.4 แสดงรูปโต๊ะ x-y

4. ชุดควบคุม ทำหน้าที่ควบคุมการทำงานของเครื่องบรรจุไข่อัตโนมัติ



รูปที่ 3.5 แสดงรูปโต๊ะ x-y

Ladder Diagram ควบคุมเครื่องบรรจุไข่อัตโนมัติ



โปรแกรมควบคุมการทำงานของเครื่องบรรจุไข่อัตโนมัติ

```

;-----
; Program          : EGG-NEST VERSION 1.12
; Description      : DC MOTOR 1-Phase Drive
;                 : X1=X-HOME,X2=Y-HOME,S1=NEST-SW,S2=END-SW
; For             : XX-XXXX-XXX
; Filename        : EGG.asm
; Assembler       : SXA51 COMPILER
;-----
;-----
; Define Port & Pin Name ( INPUT & OUTPUT )
;-----
X_SW          BIT   P1.0 ; KEY = A
Y_SW          BIT   P1.1 ; KEY = B
X_HOME       BIT   P1.2 ; KEY = C
Y_HOME       BIT   P1.3 ; KEY = D
END_SW       BIT   P1.4 ;
X_END        BIT   P1.5 ; INPUT Y END SW SENSOR
HOME_SW      BIT   P1.7 ; INPUT EGG SW DROP SENSOR (0.5 DELAY)

X_RMOTOR     BIT   P0.5 ; OUTPUT X-MOTOR ROTATE RIGHT ON/OFF
X_LMOTOR     BIT   P0.1 ; OUTPUT X-MOTOR ROTATE LEFT ON/OFF
Y_RMOTOR     BIT   P0.4 ; OUTPUT Y-MOTOR ROTATE RIGHT ON/OFF
Y_LMOTOR     BIT   P0.3 ; OUTPUT Y-MOTOR ROTATE LEFT ON/OFF
CONVEY       BIT   P0.0 ; OUTPUT RELAY ON/OFF MOTOR CONVEYOR
;-----
STOP         BIT   P3.2 ; INPUT S1 ( END RUN )
NET_SW      BIT   P3.3 ; INPUT S2 ( NET SW )
;-----
; Define User Register (RAM)
;-----
HOME         EQU   030H ; For keep Stepper (X=MIN)(Y=MIN)
X_INC       EQU   031H ; For keep Stepper X_FORWARD
X_DEC       EQU   032H ; For keep Stepper X_BACKWARD
Y_INC       EQU   033H ; For keep Stepper Y_FORWARD
Y_DEC       EQU   034H ; For keep Stepper Y_BACKWARD
END_1       EQU   035H ; For keep Stepper (X=MAX)(Y=MAX)

```

```

;-----
; Main Program. ( SET XY TABLE TO HOME )
;-----
          ORG    0000H          ; Reset Vector

          MOV    P0,#00000000B ; SET P0 AS OUTPUT Databus
          MOV    P1,#11111111B
          MOV    P2,#00000000B ; SET P2 AS OUTPUT Databus
          MOV    P3,#11111111B
;-----

INIT:      CLR    A
          MOV    A,#00D
          MOV    R0,#00D
          MOV    R1,#05D
;-----

LOOP:      ACALL DELAY_05S
          SETB   X_LMOTOR
          JNB   X_HOME,X_OFF
          SJMP  LOOP

X_OFF:     CLR    X_LMOTOR
          ACALL DELAY_HOME

LOOP1:     SETB   Y_LMOTOR          ; ON P0.3
          JNB   Y_HOME,Y_OFF

          SJMP  LOOP1

Y_OFF:     CLR    Y_LMOTOR          ; OFF P0.3
          ACALL DELAY_1S

          ACALL E_SW          ; CHECK EGG DROP SW

START:     ;
;
;-----
; MOVE X = 1 STEP INCREMENT STEP
;-----

MOV1:      SETB   CONVEY
          SETB   X_RMOTOR
          ACALL DELAY_100ms

X_SENSOR:  MOV    R0,#01
X_SENSOR1: INC    R0
          MOV    X_INC,R0

```

```

ACALL DELAY_1S

OFF:    DJNZ R1,X1_2    ; CHECK 6 EGG/COLUMN
        CLR  X_RMOTOR ; IF = 6 STOP X_RMOTOR
        ACALL E_SW    ; CHECK SW EGG DROP
        JMP  Y_MOV1   ; GOTO Y-AXIS MOVE

;-----
X1_2:
        MOV  A,X_INC
        CJNE A,#02D,X1
X1OFF:  CLR  X_RMOTOR
        ACALL E_SW    ; CHECK SW EGG DROP

X1:     SJMP  X_MOV1

;-----
; MOVE Y = 1 STEP INCREMENT ROUTINE
;-----
Y_MOV1: SETB  Y_RMOTOR
        MOV  R2,#01D  ; INC 1 ROWS
Y_SENSOR: MOV  R3,#01D
Y_SENSOR1: INC  R0
        MOV  Y_INC,R3
        ACALL DELAY_1S
OFF1:    DJNZ R2,Y1_2    ; CHECK Y-AXIS+1 MOVE
        CLR  Y_RMOTOR ; IF MOVE = 1 STOP Y_RMOTOR
        ACALL E_SW    ; CHECK SW EGG DROP
        ACALL DELAY_05S
        JMP  X_MOV2   ; GOTO X-AXIS MOVE

;-----
Y1_2:   MOV  A,Y_INC
        CJNE A,#02D,Y1
Y1OFF:  CLR  Y_RMOTOR
        ACALL E_SW    ; CHECK EGG DROP
        CLR  CONVEY
Y1:     SJMP  Y_MOV1

;-----
; MOVE X = 2 STEP DECREMENT ROUTINE
;-----
X_MOV2: MOV  R1,#05D
X_MOV2_1: CLR  X_RMOTOR
        SETB X_LMOTOR
        SETB CONVEY
X2_SENSOR: MOV  R0,#01D
X2_SENSOR1: INC  R0
        MOV  X_DEC,R0
        ACALL DELAY_1S
        DJNZ R1,X2_2    ; ADD NEW

```

```

OFF2:      ACALL DELAY_05S ; ADJUST POSITION COLUMN 2
           ACALL CLR_XH   ; CLEAR POSITION X HOME
           CLR  X_LMOTOR  ; STOP BACKWARD MOTOR
           ACALL E_SW     ; CHECK SW EGG DROP
           ACALL DELAY_05S
           JMP  Y_MOV2    ; GOTO Y AXIS+1
           SJMP X2_2

;-----
X2_2:      MOV  A,X_DEC
           CJNE A,#02D,X22
X2OFF:     CLR  X_LMOTOR
           ACALL E_SW     ; CHECK SW EGG DROP
           CLR  CONVEY
X22:       SJMP X_MOV2_1

;-----
; MOVE Y = 2 STEP INCREMENT ROUTINE
;-----
Y_MOV2:    SETB Y_RMOTOR
           SETB CONVEY
           MOV  R2,#01D ; INC 1 ROWS
Y2_SENSOR: MOV  R3,#01D
Y2_SENSOR1: INC  R0
           MOV  Y_INC,R3
           ACALL DELAY_1S
           DJNZ R2,Y2_2 ; ADD NEW
OFF3:      CLR  Y_RMOTOR ; ADD NEW
           ACALL E_SW     ; CHECK SW EGG DROP
           ACALL DELAY_05S
           JMP  X_MOV3    ; OFF3 ; ADD NEW

;-----
Y2_2:      MOV  A,Y_INC
           CJNE A,#02D,Y2
Y2OFF:     CLR  Y_RMOTOR
           ACALL E_SW     ; CHECK SW EGG DROP
           CLR  CONVEY
Y2:        SJMP Y_MOV2

;-----
; MOVE X = 3 STEP INCREMENT ROUTINE
;-----
X_MOV3:    MOV  R1,#05D
X_MOV3_1:  SETB X_RMOTOR
           CLR  X_LMOTOR
           SETB CONVEY
           ACALL DELAY_100MS
X3_SENSOR: MOV  R0,#01
X3_SENSOR1: INC  R0
           MOV  X_INC,R0
           ACALL DELAY_1S

```



```

CLR X_LMOTOR ; ADD NEW
ACALL E_SW ; CHECK SW EGG DROP
JMP Y_MOV4 ; STOP X = 4 HERE
SJMP X4_2
;-----
X4_2: MOV A,X_DEC
CJNE A,#02D,X4
X4OFF: CLR X_LMOTOR
ACALL E_SW ; CHECK EGG DROP
CLR CONVEY
X4: SJMP X_MOV4_1
;-----
; MOVE Y = 4 STEP INCREMENT ROUTINE
;-----
Y_MOV4: SETB Y_RMOTOR
SETB CONVEY
MOV R2,#01D ; INC 1 ROWS
Y4_SENSOR: MOV R3,#01D
Y4_SENSOR1: INC R0
MOV Y_INC,R3
ACALL DELAY_1S
DJNZ R2,Y4_2 ; DEC TO 0
OFF7: CLR Y_RMOTOR ; OFF Y_RMOTOR
ACALL E_SW ; CHECK SW EGG DROP
JMP X_MOV5 ; MOVE TO ANOTHER COLUMN
;-----
Y4_2: MOV A,Y_INC
CJNE A,#02D,Y4
Y4OFF: CLR Y_RMOTOR
ACALL E_SW ; CHECK EGG DROP
CLR CONVEY
Y4: SJMP Y_MOV4
;-----
; MOVE X = 5 STEP DECREMENT ROUTINE
;-----
X_MOV5: MOV R1,#05D
X_MOV5_1: CLR X_LMOTOR
SETB X_RMOTOR
SETB CONVEY
X5_SENSOR: MOV R0,#01D
X5_SENSOR1: INC R0
MOV X_DEC,R0
ACALL DELAY_1S
DJNZ R1,X5_2 ; ADD NEW
ACALL DELAY_05S
OFF8: CLR X_RMOTOR ; ADD NEW
ACALL E_SW ; CHECK SW EGG DROP
CLR CONVEY

```

```

JMP END_SHOOT ;STOP X = 5 EJECT NET TO END
SJMP X5_2
;-----
X5_2:    MOV    A,X_DEC
        CJNE  A,#02D,X5
X5OFF:  CLR    X_RMOTOR
        ACALL E_SW      ; CHECK SW EGG DROP
        CLR   CONVEY
X5:     SJMP  X_MOV5_1
;-----
;      END OF SHOOT EGG & EJECT NET TO END SW ROUTINE
;-----

END_SHOOT:
        SETB  X_RMOTOR
        ACALL DELAY_1S
        SETB  Y_RMOTOR
        ACALL DELAY_1S

CHK:    CLR    X_RMOTOR
        ACALL DELAY_100MS
        CLR   Y_RMOTOR
        ACALL DELAY_100MS
        JNB  NET_SW,BACK
        JMP  CHK

BACK:   MOV    R0,#00D
        MOV   R1,#05D
        JMP   LOOP

E_SW:   SETB  CONVEY
        JNB  X_SW,OUT  ; JNB EGG_SW,OUT
        SJMP E_SW

OUT:    RET

;-----
CLR_XH: SETB  X_LMOTOR
        JNB  X_HOME,ADJ_HOME
        JMP  CLR_XH
ADJ_HOME: CLR  X_LMOTOR
        RET
;-----

```

```

;-----
CLR_XE:      SETB  X_RMOTOR
             JNB   X_END,ADJ_END
             JMP   CLR_XE
ADJ_END:     CLR   X_RMOTOR
             RET
;-----
;-----

```

```

; Dummy Delay time 100mS ( 0.1 SEC,1-SEC,2-SEC,5-SEC ) AT 11.0529 MHZ
;-----

```

```

DELAY_100ms:  MOV   R7,#10      ; Do 10 times
DELAY_100ms_1: MOV   R6,#0E6H    ; Each loop = 1 ms
DELAY_100ms_2: NOP
             NOP
             DJNZ  R6,DELAY_100ms_2
             DJNZ  R7,DELAY_100ms_1
             RET

```

```

DELAY_05S:    MOV   R5,#30      ;#30 EQUAL TIME 0.3 SEC
DELAY_05S_1:  ACALL DELAY_100MS
             DJNZ  R5,DELAY_05S_1
             RET

```

```

DELAY_1S:     MOV   R5,#135     ;#130 EQUAL TIME 1.3 SEC
DELAY_1S_1:   ACALL DELAY_100MS
             DJNZ  R5,DELAY_1S_1
             RET

```

```

DELAY_2X:     MOV   R5,#100     ;#130 EQUAL TIME 1.3 SEC
DELAY_2X_1:   ACALL DELAY_100MS
             DJNZ  R5,DELAY_2X_1
             RET

```

```

DELAY_5S:     MOV   R5,#200     ;#200 EQUAL TIME 2 SEC
DELAY_5S_1:   ACALL DELAY_100MS
             DJNZ  R5,DELAY_5S_1
             RET

```

```

DELAY_2S:     MOV   R5,#2
DELAY_2S_1:   ACALL DELAY_100MS
             DJNZ  R5,DELAY_2S_1
             RET

```

```

DELAY_10S:    MOV   R4,#900     ; Do 90 times
DELAY_10S_1:  ACALL DELAY_100MS ; Each loop = 1 ms
             DJNZ  R4,DELAY_10S_1
             RET

```

```
DELAY_HOME:      MOV  R5,#50
DELAY_HOME1:     ACALL DELAY_100MS
                 DJNZ  R5, DELAY_HOME1
                 RET
                 END
```