

```
#include "stepper.h"
```

```
#define DELAY1 975
#define DELAY2 1500
#define DELAY3 1875
```

```
const struct State{
    unsigned char Out;           // Output to Port T
    unsigned short delay;
    const struct State *Next[8]; // Next state if input=0,1,2,3
};
```

```
typedef const struct State StateType;
typedef StateType * StatePtr;
```

```
#define Normal 5 &fsm[0]
#define Normal 6 &fsm[1]
#define Normal 10 &fsm[2]
#define Normal 9 &fsm[3]
#define One5 &fsm[4]
#define One6 &fsm[5]
#define One10 &fsm[6]
#define One9 &fsm[7]
#define Vi b1 &fsm[8]
#define Vi b2 &fsm[9]
#define Vi b3 &fsm[10]
#define Vi b4 &fsm[11]
#define Vi b5 &fsm[12]
#define Vi b6 &fsm[13]
#define Vi b7 &fsm[14]
#define Vi b8 &fsm[15]
#define Vi b9 &fsm[16]
#define Vi b10 &fsm[17]
#define Vi b11 &fsm[18]
#define Vi b12 &fsm[19]
#define Vi b13 &fsm[20]
#define Vi b14 &fsm[21]
```

```
StateType fsm[22]={
    {0x05, DELAY1, { Normal 5, Vi b1, One5, Normal 5, Normal 6, Normal 5, Normal 9,
Normal 5}}, // Normal 5
    {0x06, DELAY1, { Normal 6, Vi b2, One6, Normal 6, Normal 10, Normal 6, Normal 5,
Normal 6}}, // Normal 6
    {0x0A, DELAY1, {Normal 10, Vi b3, One10, Normal 10, Normal 9, Normal 10, Normal 6,
Normal 10}}, // Normal 10
    {0x09, DELAY1, { Normal 9, Vi b4, One9, Normal 9, Normal 5, Normal 9, Normal 10,
Normal 9}}, // Normal 9

    {0x05, 50, { Normal 6, Vi b1, One5, One5, Normal 6, One5, Normal 9,
One5}}, // One5
    {0x06, 50, {Normal 10, Vi b2, One6, One6, Normal 10, One6, Normal 5,
One6}}, // One6
    {0x0A, 50, { Normal 9, Vi b3, One10, One10, Normal 9, One10, Normal 6,
One10}}, // One10
    {0x09, 50, { Normal 5, Vi b4, One9, One9, Normal 5, One9, Normal 10,
One9}}, // One9

    {0x06, DELAY3, { Vi b1, Vi b2, One6, Vi b1, Normal 10, Vi b1, Normal 5,
Vi b1}}, // Vi b1
    {0x0A, DELAY1, { Vi b2, Vi b3, One10, Vi b2, Normal 9, Vi b2, Normal 6,
Vi b2}}, // Vi b2
    {0x09, DELAY1, { Vi b3, Vi b4, One9, Vi b3, Normal 5, Vi b3, Normal 10,
Vi b3}}, // Vi b3
```

```

stepper.c
{0x05, DELAY1, { Vi b4, Vi b5, One5, Vi b4, Normal 6, Vi b4, Normal 9,
Vi b4}}, // Vi b4
{0x06, DELAY3, { Vi b5, Vi b6, One6, Vi b5, Normal 10, Vi b5, Normal 5,
Vi b5}}, // Vi b5
{0x0A, DELAY2, { Vi b6, Vi b7, One10, Vi b6, Normal 9, Vi b6, Normal 6,
Vi b6}}, // Vi b6
{0x09, DELAY2, { Vi b7, Vi b8, One9, Vi b7, Normal 5, Vi b7, Normal 10,
Vi b7}}, // Vi b7

{0x0A, DELAY3, { Vi b8, Vi b9, One10, Vi b8, Normal 9, Vi b8, Normal 6,
Vi b8}}, // Vi b8
{0x06, DELAY1, { Vi b9, Vi b10, One6, Vi b9, Normal 10, Vi b9, Normal 5,
Vi b9}}, // Vi b9
{0x05, DELAY1, { Vi b10, Vi b11, One5, Vi b10, Normal 6, Vi b10, Normal 9,
Vi b10}}, // Vi b10
{0x09, DELAY1, { Vi b11, Vi b12, One9, Vi b11, Normal 5, Vi b11, Normal 10,
Vi b11}}, // Vi b11
{0x0A, DELAY3, { Vi b12, Vi b13, One10, Vi b12, Normal 9, Vi b12, Normal 6,
Vi b12}}, // Vi b12
{0x06, DELAY2, { Vi b13, Vi b14, One6, Vi b13, Normal 10, Vi b13, Normal 5,
Vi b13}}, // Vi b13
{0x05, DELAY2, { Vi b14, Vi b1, One5, Vi b14, Normal 6, Vi b14, Normal 9,
Vi b14}}, // Vi b14
};
StatePtr Pt; // Current State

//-----OC_Init0-----
// arm output compare 0 for 1 Hz periodic interrupt
// also enables timer to 16 us period
// Input: none
// Output: none
void OC_Init0(void){
    Pt = Normal 5;
    DDRP = 0x8F;
    PTP = 0x80;
    TIOS = 0x01; // activate TC0 as output compare
    TIE = 0x01; // arm OCO
    TSCR1 = 0x80; // Enable TCNT, 24MHz boot mode, 8MHz in run mode
    TSCR2 = 0x07; // divide by 128 TCNT prescale, TOI disarm, sets period to 5.33us
    PACTL = 0; // timer prescale used for TCNT
    TC0 = TCNT+50; // first interrupt right away
}

//-----Output Compare Interrupt-----
// output, set up delay, go to next state
//
interrupt 8 void TOC0handler(void){ // executes at 1/delay Hz
    TFLG1 = 0x01; // acknowledge OCO
    PTP = (PTP&0xF0) + Pt->Out; // output to stepper motor
    TC0 = TC0 + Pt->delay; // interrupts again after state delay
    Pt = Pt->Next[(PTP&0x70) >> 4]; // change to next state
}

```