```
#include <hidef.h>      /* common defines and macros */
#include "derivative.h"      /* derivative-specific definitions */

#include "PLL.h"
#include "stepper.h"
#include "switch.h"

void main(void) {
  PLL_Init();
  switch_Init();
  OC_Init0();

  asm cli

  for(;;) {} // Everything important runs in background
}
```

```
#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 Normal5  &fsm[0]
#define Normal6  &fsm[1]
#define Normal10 &fsm[2]
#define Normal9  &fsm[3]
#define One5     &fsm[4]
#define One6     &fsm[5]
#define One10    &fsm[6]
#define One9     &fsm[7]
#define Vib1     &fsm[8]
#define Vib2     &fsm[9]
#define Vib3     &fsm[10]
#define Vib4     &fsm[11]
#define Vib5     &fsm[12]
#define Vib6     &fsm[13]
#define Vib7     &fsm[14]
#define Vib8     &fsm[15]
#define Vib9     &fsm[16]
#define Vib10    &fsm[17]
#define Vib11    &fsm[18]
#define Vib12    &fsm[19]
#define Vib13    &fsm[20]
#define Vib14    &fsm[21]

StateType fsm[22]={
  {0x05, DELAY1, { Normal5,  Vib1,  One5,  Normal5,  Normal6,  Normal5,  Normal9,
Normal5}}, // Normal5
  {0x06, DELAY1, { Normal6,  Vib2,  One6,  Normal6, Normal10,  Normal6,  Normal5,
Normal6}}, // Normal6
  {0x0A, DELAY1, {Normal10,  Vib3, One10, Normal10,  Normal9, Normal10,  Normal6,
Normal10}}, // Normal10
  {0x09, DELAY1, { Normal9,  Vib4,  One9,  Normal9,  Normal5,  Normal9, Normal10,
Normal9}}, // Normal9

  {0x05,     50, { Normal6,  Vib1,  One5,     One5,  Normal6,     One5,  Normal9,
 One5}}, // One5
  {0x06,     50, {Normal10,  Vib2,  One6,     One6, Normal10,     One6,  Normal5,
 One6}}, // One6
  {0x0A,     50, { Normal9,  Vib3, One10,    One10,  Normal9,    One10,  Normal6,
One10}}, // One10
  {0x09,     50, { Normal5,  Vib4,  One9,     One9,  Normal5,     One9, Normal10,
 One9}}, // One9

  {0x06, DELAY3, {    Vib1,  Vib2,  One6,     Vib1, Normal10,     Vib1,  Normal5,
 Vib1}}, // Vib1
  {0x0A, DELAY1, {    Vib2,  Vib3, One10,     Vib2,  Normal9,     Vib2,  Normal6,
 Vib2}}, // Vib2
  {0x09, DELAY1, {    Vib3,  Vib4,  One9,     Vib3,  Normal5,     Vib3, Normal10,
 Vib3}}, // Vib3
```

```
  {0x05, DELAY1, {    Vib4,  Vib5,  One5,     Vib4,  Normal6,     Vib4,  Normal9,
 Vib4}}, // Vib4
  {0x06, DELAY3, {    Vib5,  Vib6,  One6,     Vib5, Normal10,     Vib5,  Normal5,
 Vib5}}, // Vib5
  {0x0A, DELAY2, {    Vib6,  Vib7, One10,     Vib6,  Normal9,     Vib6,  Normal6,
 Vib6}}, // Vib6
  {0x09, DELAY2, {    Vib7,  Vib8,  One9,     Vib7,  Normal5,     Vib7, Normal10,
 Vib7}}, // Vib7

  {0x0A, DELAY3, {    Vib8,  Vib9, One10,     Vib8,  Normal9,     Vib8,  Normal6,
 Vib8}}, // Vib8
  {0x06, DELAY1, {    Vib9, Vib10,  One6,     Vib9, Normal10,     Vib9,  Normal5,
 Vib9}}, // Vib9
  {0x05, DELAY1, {   Vib10, Vib11,  One5,    Vib10,  Normal6,    Vib10,  Normal9,
Vib10}}, // Vib10
  {0x09, DELAY1, {   Vib11, Vib12,  One9,    Vib11,  Normal5,    Vib11, Normal10,
Vib11}}, // Vib11
  {0x0A, DELAY3, {   Vib12, Vib13, One10,    Vib12,  Normal9,    Vib12,  Normal6,
Vib12}}, // Vib12
  {0x06, DELAY2, {   Vib13, Vib14,  One6,    Vib13, Normal10,    Vib13,  Normal5,
Vib13}}, // Vib13
  {0x05, DELAY2, {   Vib14,  Vib1,  One5,    Vib14,  Normal6,    Vib14,  Normal9,
Vib14}}, // Vib14
};
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 = Normal5;
  DDRP |= 0x8F;
  PTP  |= 0x80;
  TIOS |= 0x01;   // activate TC0 as output compare
  TIE  |= 0x01;   // arm OC0
  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 OC0
  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
}
```

```
#include <hidef.h>      /* common defines and macros */
#include "derivative.h"      /* derivative-specific definitions */


//---------------------OC_Init0---------------------
// arm output compare 0 for 4 Hz periodic interrupt
// also enables timer to 16 us period
// Input: none
// Output: none
void OC_Init0(void);
```

```
#include "switch.h"

//---------------------switch_Init---------------------
// Set PP6-4 as inputs
// Input: none
// Output: none
void switch_Init() {
  DDRP &= ~0x70;
}
```

```
#include <hidef.h>      /* common defines and macros */
#include "derivative.h"      /* derivative-specific definitions */

//---------------------switch_Init---------------------
// Set PP6-4 as inputs
// Input: none
// Output: none
void switch_Init(void);
```