```c
/*
 * Academic License - for use in teaching, academic research, and meeting
 * course requirements at degree granting institutions only.  Not for
 * government, commercial, or other organizational use.
 *
 * File: ert_main.c
 *
 * Code generated for Simulink model 'MPDPC_DSP'.
 *
 * Model version                  : 1.864
 * Simulink Coder version         : 9.0 (R2018b) 24-May-2018
 * C/C++ source code generated on : Fri Jan  8 10:43:21 2021
 *
 * Target selection: ert.tlc
 * Embedded hardware selection: Texas Instruments->C2000
 * Code generation objectives: Unspecified
 * Validation result: Not run
 */

#include "MPDPC_DSP.h"
#include "rtwtypes.h"

volatile int IsrOverrun = 0;
static boolean_T OverrunFlag = 0;
void rt_OneStep(void)
{
  /* Check for overrun. Protect OverrunFlag against preemption */
  if (OverrunFlag++) {
    IsrOverrun = 1;
    OverrunFlag--;
    return;
  }

  enableTimer0Interrupt();
  MPDPC_DSP_step();

  /* Get model outputs here */
  disableTimer0Interrupt();
  OverrunFlag--;
}

volatile boolean_T stopRequested = false;
int main(void)
{
  volatile boolean_T runModel = true;
  float modelBaseRate = 5.0E-5;
  float systemClock = 150;
  c2000_flash_init();
  init_board();

#ifdef MW_EXEC_PROFILER_ON

  config_profilerTimer();

#endif
```

```
  ;
  rtmSetErrorStatus(MPDPC_DSP_M, 0);
  MPDPC_DSP_initialize();
  configureTimer0(modelBaseRate, systemClock);
  runModel =
    rtmGetErrorStatus(MPDPC_DSP_M) == (NULL);
  enableTimer0Interrupt();
  globalInterruptEnable();
  while (runModel) {
    stopRequested = !(
                      rtmGetErrorStatus(MPDPC_DSP_M) == (NULL));
    runModel = !(stopRequested);
  }

  /* Disable rt_OneStep() here */

  /* Terminate model */
  MPDPC_DSP_terminate();
  globalInterruptDisable();
  return 0;
}

/*
 * File trailer for generated code.
 *
 * [EOF]
 */
```