File: C:\Users\Michael\Documents\Courses\15ENGR120\Assignments\Assignment_1\Traf

```
// Traffic Light Skeleton Code.

// Enumerated type of system states
typedef enum T_FSMState {
  GreenLight1RedLight2=0,
  YellowLight1RedLight2,
  RedLight1GreenLight2,
  RedLight1YellowLight2,
  RedLight1RedLight2,
  FourWayStop
};

// Enumerated type of light control settings
typedef enum T_LightControl {
  RedLightOn=0,
  YellowLightOn,
  GreenLightOn,
  FlashingRed
};

// Light durations in multiples of 100 miliseconds
const int YELLOW_LIGHT_TIME = 50;   // Yellow lights last for 5.0 seconds.
const int RED_LIGHT_TIME1   = 200;  // Minimum duration for red lights
//   on direction 1 is 20.0 seconds.
const int RED_LIGHT_TIME2   = 300;  // Minimum duration for red lights
//   on direction 2 is 30.0 seconds.

// We are using timer T1 for controlling the lights.
#define LIGHTTIMER T1

// Declaration of function (full definitions are later)
// State processing
T_FSMState ProcessGreenLight1RedLight2();
T_FSMState ProcessYellowLight1RedLight2();
T_FSMState ProcessRedLight1RedLight2();
T_FSMState ProcessRedLight1GreenLight2();
T_FSMState ProcessRedLight1YellowLight2();
T_FSMState ProcessFourWayStop();

// Input checking functions
// Full code not provided.
bool CheckTrafficDirection1();  // Returns true if incoming traffic
//  from direction 1, otherwise returns false
bool CheckTrafficDirection2();  // Returns true if incoming traffic
//  from direction 2, otherwise returns false
bool CheckIntersectionClear();  // Returns true if intersection is clear,
//   otherwise returns false.

bool CheckAfterHours();          // Returns true if after 10PM or before 6AM,
//   otherwise returns false.

// SetLights sets the light in each direction.
void SetLights( T_LightControl controlDirection1,
T_LightControl controlDirection2);

// Process green light direction 1, and red light direction 2 state.
T_FSMState ProcessGreenLight1RedLight2() {
  // Set the traffic lights
  SetLights(GreenLightOn,RedLightOn);

  // Check the conditions for state change.
  if ( time1(LIGHTTIMER) > RED_LIGHT_TIME2 && CheckTrafficDirection2() ) {
    return YellowLight1RedLight2;
    } else {
    return GreenLight1RedLight2;
  }
}

// Process yellow light direction 1, and red light direction 2 state.
T_FSMState ProcessYellowLight1RedLight2() {

  // Set the traffic lights
  SetLights(YellowLightOn,RedLightOn);

  // Check the conditions for state change.
  if ( time1(LIGHTTIMER) > YELLOW_LIGHT_TIME && CheckIntersectionClear() ) {
    return RedLight1GreenLight2;
    } else {
    return YellowLight1RedLight2;
  }
}

// Process red light direction 1, and green light direction 2 state.
T_FSMState ProcessRedLight1GreenLight2() {

  // Set the traffic lights
  SetLights(RedLightOn,GreenLightOn);

  // Check the conditions for state change.
  if ( time1(LIGHTTIMER) > RED_LIGHT_TIME1 && CheckTrafficDirection1() ) {
    return RedLight1YellowLight2;
    } else {
    return RedLight1GreenLight2;
  }

}

// Process red light direction 1, and yellow light direction 2 state.
T_FSMState ProcessRedLight1YellowLight2() {

  // Set the traffic lights
  SetLights(RedLightOn,YellowLightOn);

  // Check the conditions for state change.
  if ( time1(LIGHTTIMER) > YELLOW_LIGHT_TIME && CheckIntersectionClear() ) {

    if ( CheckAfterHours() ) {
      return RedLight1RedLight2;
    } else {
      return GreenLight1RedLight2;
    }

  } else {
    return RedLight1YellowLight2;
  }

}

// Process red light in both directions state.
T_FSMState ProcessRedLight1RedLight2() {

  // Set the traffic lights
  SetLights(RedLightOn,RedLightOn);

  // Check the conditions for state change.
  if ( CheckAfterHours() ) {
    return FourWayStop;
  }
  else if ( CheckIntersectionClear() ) {
    return GreenLight1RedLight2;
  } else {
    return RedLight1RedLight2;
  }
}

// Process red lights as a four way stop.
T_FSMState ProcessFourWayStop() {

  // Flash all lights red when in this state.
  SetLights(FlashingRed,FlashingRed);

  // Check if we have entered regular business hours.
  if ( ! CheckAfterHours() ) {
    // We are not after hours, transition to all lights red.
    return RedLight1RedLight2;
    } else {
    // Remain in four way stop mode.
    return FourWayStop;
  }

}

// This is the main code.
task main() {

  // Start in the state for red lights in all directions.
  T_FSMState currentState = RedLight1RedLight2;
  T_FSMState nextState;

  while( true ) {

    switch(currentState) {
    case GreenLight1RedLight2:
      nextState = ProcessGreenLight1RedLight2();
      break;
    case YellowLight1RedLight2:
      nextState = ProcessYellowLight1RedLight2();
      break;
    case RedLight1GreenLight2:
      nextState = ProcessRedLight1GreenLight2();
      break;
    case RedLight1YellowLight2:
      nextState = ProcessRedLight1YellowLight2();
      break;
    case RedLight1RedLight2:
      nextState = ProcessRedLight1RedLight2();
      break;
    case FourWayStop:
      nextState = ProcessFourWayStop();
      break;
    default:
      // Full code would log error and change to failsafe mode
      // Putting redlights in both directions is the safest mode we have.
      nextState = ProcessRedLight1RedLight2();
    } // switch(currentState)

    // Switch to computed next state
    if ( currentState != nextState ) {
      // If we have changed state, reset the timer.
      clearTimer(LIGHTTIMER);
    }
    currentState = nextState;

  } // while(true)
}
```