```
#pragma config(Hubs,  S1, HTMotor,  HTServo,  HTMotor,  none)
#pragma config(Sensor, S1,     ,             sensorI2CMuxController)
#pragma config(Sensor, S2,     touchtop,     sensorTouch)
#pragma config(Sensor, S3,     touchbottom,  sensorTouch)
#pragma config(Sensor, S4,     HTSMUX,       sensorI2CCustom)
#pragma config(Motor,  mtr_S1_C1_1,     FR,          tmotorTetrix, openLoop)
#pragma config(Motor,  mtr_S1_C1_2,     FL,          tmotorTetrix, openLoop, reversed)
#pragma config(Motor,  mtr_S1_C3_1,     BR,          tmotorTetrix, openLoop)
#pragma config(Motor,  mtr_S1_C3_2,     BL,          tmotorTetrix, openLoop, reversed)
#pragma config(Servo,  srvo_S1_C2_1,    servo1,           tServoStandard)
#pragma config(Servo,  srvo_S1_C2_2,    servo2,           tServoStandard)
#pragma config(Servo,  srvo_S1_C2_3,    servo3,           tServoStandard)
#pragma config(Servo,  srvo_S1_C2_4,    servo4,           tServoNone)
#pragma config(Servo,  srvo_S1_C2_5,    servo5,           tServoNone)
#pragma config(Servo,  srvo_S1_C2_6,    servo6,           tServoNone)
//*!!Code automatically generated by 'ROBOTC' configuration wizard               !!*//

////////////////////////////////////////////////////////////////////////////////////////////////////
//
//                    Tele-Operation Mode Code Template
//
// This file contains a template for simplified creation of an tele-op program for an FTC
// competition
//
// You need to customize two functions with code unique to your specific robot.
//
////////////////////////////////////////////////////////////////////////////////////////////////////

#include "JoystickDriver.c"  //Include file to "handle" the Bluetooth messages.
#include "hitechnic-sensormux.h" //Driver for Sensor Multiplexer by HiTechnic
#include "hitechnic-irseeker-v2.h" //Driver for IRSeeker
#include "drivers/lego-ultrasound.h" //Driver for Ultrasonic Sensor By Lego
#include "lego-touch.h" // Driver for both Touch Sensors from Lego

////////////////////////////////////////////////////////////////////////////////////////////////////
//
//                          initializeRobot
//
// Prior to the start of tele-op mode, you may want to perform some initialization on your robot
// and the variables within your program.
//
// In most cases, you may not have to add any code to this function and it will remain "empty".
//
////////////////////////////////////////////////////////////////////////////////////////////////////

//void initializeRobot()
//{
```

```
//   //Place code here to initialize servos to starting positions.
//   //Sensors are automatically configured and setup by ROBOTC. They may need a brief time to stabilize.

//  return;
//}


////////////////////////////////////////////////////////////////////////////////////////////////
//
//                                    Main Task
//
// The following is the main code for the tele-op robot operation. Customize as appropriate for
// your specific robot.
//
// Game controller / joystick information is sent periodically (about every 50 milliseconds) from
// the FMS (Field Management System) to the robot. Most tele-op programs will follow the following
// logic:
//   1. Loop forever repeating the following actions:
//   2. Get the latest game controller / joystick settings that have been received from the PC.
//   3. Perform appropriate actions based on the joystick + buttons settings. This is usually a
//      simple action:
//      *  Joystick values are usually directly translated into power levels for a motor or
//         position of a servo.
//      *  Buttons are usually used to start/stop a motor or cause a servo to move to a specific
//         position.
//   4. Repeat the loop.
//
// Your program needs to continuously loop because you need to continuously respond to changes in
// the game controller settings.
//
// At the end of the tele-op period, the FMS will autonmatically abort (stop) execution of the program.
//
////////////////////////////////////////////////////////////////////////////////////////////////

task main()
{
  //initializeRobot();

  //waitForStart();   // wait for start of tele-op phase
  while (true)
            {
                      int threshold = 5;          /* Int 'threshold' will allow us to ignore low      */
                      int thresholdX = 50;                      /* readings that keep our robot in perpetual motion. */

                        getJoystickSettings(joystick);
                        if(abs(joystick.joy1_y2) > threshold)   // If the right analog stick's Y-axis readings are either above or
below the threshold:
```

```
                {
                  motor[FL] = joystick.joy1_y2/3;       // Motor D is assigned a power level equal to the right
analog stick's Y-axis reading.
                  motor[BL] = joystick.joy1_y2/3;
                }
          else                        // Else if the readings are within the threshold:
                {
                  motor[FL] = 0;                  // Motor D is stopped with a power level of 0.
                  motor[BL] = 0;
                }


          if(abs(joystick.joy1_y1) > threshold)   // If the left analog stick's Y-axis readings are either above or
below the threshold:
                {
                  motor[FR] = joystick.joy1_y1/3;       // Motor E is assigned a power level equal to the left
analog stick's Y-axis reading.
                  motor[BR] = joystick.joy1_y1/3;
                }
          else                        // Else if the readings are within the threshold:
                {
                  motor[FR] = 0;                  // Motor E is stopped with a power level of 0.
                  motor[BR] = 0;
                }


          if((abs(joystick.joy1_x1) > thresholdX))
                {
                  if((joystick.joy1_x2) < 0 && (abs(joystick.joy1_x2) > thresholdX))
                  {
                        //(joystick.joy1_x1 = joystick.joy1_x2);
                        //Left
                        //motor[FL] = (joystick.joy1_x1)/3;
                        //motor[FR] = (joystick.joy1_x1/-3);
                        //motor[BL] = (joystick.joy1_x1/-3);
                        //motor[BR] = (joystick.joy1_x1)/3;
                  motor[FL] = 80;
                                          motor[FR] = -80;
                                          motor[BL] = -80;
                                          motor[BR] = 80;
                        }
                        if((joystick.joy1_x2) > 0 && (abs(joystick.joy1_x2) > thresholdX))
                        {
                                  //(joystick.joy1_x1 = joystick.joy1_x2);
                                  //Right
                                  //motor[FL] = (joystick.joy1_x1*-1)/3;
                    //    motor[FR] = (joystick.joy1_x1)/3;
```

```
            //    motor[BL] = (joystick.joy1_x1)/3;
            //    motor[BR] = (joystick.joy1_x1*-1)/3;
                              motor[FL] = -60;
                                                motor[FR] = 60;
                                                motor[BL] = 60;
                                                motor[BR] = -60;
                        }
                }



if(joy1Btn(6))        // If Button 5 is pressed:
        {
         servo[servo1] = 210;     // Raise Servo 1 to position 38.
        }

if(joy1Btn(5))        // If Button 6 is pressed:
        {
         servo[servo1] = 90;      // Lower Servo 1 to position 165.
        }

if(joy1Btn(2))        // If Button 2 is pressed:
        {
         servo[servo2] = 0;     // Raise Servo 2 to position 15.
        }

if(joy1Btn(4))        // If Button 4 is pressed:
        {
         servo[servo2] = 225;      // Lower Servo 2 to position 215.
        }
                              if(joy1Btn(1))        // If Button 2 is pressed:
        {
         servo[servo3] = 0;     // Raise Servo 2 to position 15.
        }

if(joy1Btn(3))        // If Button 4 is pressed:
        {
         servo[servo3] = 255;      // Lower Servo 2 to position 215.
        }
       if(joy1Btn(7))                                    // If Button 7 is pressed:
      {
                motor[motorB] = 180;                     // Move the flag motor at 100% backward
      }
if(joy1Btn(8))                                      // If Button 8 is pressed:
      {
                motor[motorB] = 100;                     // Move the flag motor at 100% forward
      }
```

```
                if(joy1Btn(7)== 0 && joy1Btn(8) == 0)          // If NEITHER Button 7 or 8 is pressed:
                {
                        motor[motorB] = 0;                      // Stop MOTOR B
                }
                        //if(joy1Btn(11)== 1 && joy1Btn(12) == 1)
                        //{
                                //  motor[motorB] = 0;
                                //  motor[FL] = 0;
                                //  motor[FR] = 0;
                                //  motor[BL] = 0;
                                //}

                //      if(joystick.joy1_TopHat == 2)           //If the TOPHAT Switch is pressed UP AND the
TOP touch sensor is NOT being pressed:
        //      {
                //      motor[FL] = -20;
                //      motor[FR] = 20;
                //      motor[BL] = 20;
                //      motor[BR] = -20;
                //      wait1Msec(2000);
        //              }
                //if(joystick.joy1_TopHat == 6)             //If the TOPHAT Switch is pressed UP AND the TOP
touch sensor is NOT being pressed:
        //      {
                //              //Right
                //              motor[FL] = 20;
                //      motor[FR] = -20;
                //      motor[BL] = -20;
                //      motor[BR] = 20;
                //      wait1Msec(2000);
        //              }


        }
}
```