```c
#pragma config(Hubs,  S1, HTMotor,  HTMotor,  HTServo,  none)
#pragma config(Sensor, S1,        ,              sensorI2CMuxController)
#pragma config(Sensor, S2,      SMUX,            sensorI2CCustom)
#pragma config(Sensor, S3,      sGyro,           sensorI2CHiTechnicGyro)
#pragma config(Sensor, S4,      sLiftStop,       sensorTouch)
#pragma config(Motor,  motorA,           mFlagRaise1,   tmotorNXT, openLoop)
#pragma config(Motor,  motorB,           mFlagRaise2,   tmotorNXT, openLoop)
#pragma config(Motor,  motorC,           mBlockStop,    tmotorNXT, openLoop, reversed)
#pragma config(Motor,  mtr_S1_C1_1,      mDriveLeft,    tmotorTetrix, openLoop, encoder)
#pragma config(Motor,  mtr_S1_C1_2,      mDriveRight,   tmotorTetrix, openLoop, reversed, encoder)
#pragma config(Motor,  mtr_S1_C2_1,      mBsConveyor,   tmotorTetrix, openLoop, reversed)
#pragma config(Motor,  mtr_S1_C2_2,      mBsAngle,      tmotorTetrix, openLoop, reversed, encoder)
#pragma config(Servo,  srvo_S1_C3_1,    rConveyorTight,      tServoStandard)
#pragma config(Servo,  srvo_S1_C3_2,    rBlockDropLeft,      tServoStandard)
#pragma config(Servo,  srvo_S1_C3_3,    rBlockDropRight,     tServoStandard)
#pragma config(Servo,  srvo_S1_C3_4,    servo4,              tServoNone)
#pragma config(Servo,  srvo_S1_C3_5,    servo5,              tServoNone)
#pragma config(Servo,  srvo_S1_C3_6,    servo6,              tServoNone)
//*!!Code automatically generated by 'ROBOTC' configuration wizard          !!*//

#include "JoystickDriver.c"
#include "autonomous-includes/autonomousTasks.h"
#include "autonomous-includes/autoMenu.h"
#include "autonomous-includes/powerStackerSplash.h"

// 1 wheel rotation = 1350 ticks

task main()
{
  //doPowerstackerSplash();
  StartTask(runMenuOffensive);
  initializeRobot();

  waitForStart();
  printMenuChoices();
  StartTask(showDebugInfo);
  StartTask(getSmux);
  startEncoderPos = nMotorEncoder[mDriveRight];
  // Is there a delay?
  if(delay > 0){
    writeDebugStreamLine("Wait detected");
    wait10Msec(delay * 100);    // Wait the amount of time given by the delay
    writeDebugStreamLine("Wait over");
  }
```

```c
  // Are we going to place the block?
  //int basketPosition = 0;
  if(doIr){
    findIrIncremental();
  }

  PlaySound(soundBeepBeep);
  wait10Msec(100);
  writeDebugStreamLine("\n----- END -----\n");

}
```