#pragma config(Motor, port2, LftDriveMotorA, tmotorVex393TurboSpeed\_MC29, openLoop)

#pragma config(Motor, port1, LftDriveMotorB, tmotorVex393TurboSpeed\_MC29, openLoop)

#pragma config(Motor, port4, RgtDriveMotorA, tmotorVex393TurboSpeed\_MC29, openLoop)

#pragma config(Motor, port5, RgtDriveMotorB, tmotorVex393TurboSpeed\_MC29, openLoop)

#pragma config(Motor, port6, MobileGoalA, tmotorVex393TurboSpeed\_MC29, openLoop)

#pragma config(Motor, port7, MobileGoalB, tmotorVex393TurboSpeed\_MC29, openLoop)

//\*!!Code automatically generated by 'ROBOTC' configuration wizard !!\*//

/\*---------------------------------------------------------------------------\*/

/\* \*/

/\* Description: Competition template for VEX EDR \*/

/\* \*/

/\*---------------------------------------------------------------------------\*/

// This code is for the VEX cortex platform

#pragma platform(VEX2)

// Select Download method as "competition"

#pragma competitionControl(Competition)

//Main competition background code...do not modify!

#include "Vex\_Competition\_Includes.c"

/\*---------------------------------------------------------------------------\*/

/\* Pre-Autonomous Functions \*/

/\* \*/

/\* You may want to perform some actions before the competition starts. \*/

/\* Do them in the following function. You must return from this function \*/

/\* or the autonomous and usercontrol tasks will not be started. This \*/

/\* function is only called once after the cortex has been powered on and \*/

/\* not every time that the robot is disabled. \*/

/\*---------------------------------------------------------------------------\*/

void pre\_auton()

{

 // Set bStopTasksBetweenModes to false if you want to keep user created tasks

 // running between Autonomous and Driver controlled modes. You will need to

 // manage all user created tasks if set to false.

 bStopTasksBetweenModes = true;

 // Set bDisplayCompetitionStatusOnLcd to false if you don't want the LCD

 // used by the competition include file, for example, you might want

 // to display your team name on the LCD in this function.

 // bDisplayCompetitionStatusOnLcd = false;

 // All activities that occur before the competition starts

 // Example: clearing encoders, setting servo positions, ...

}

/\*---------------------------------------------------------------------------\*/

/\* \*/

/\* Autonomous Task \*/

/\* \*/

/\* This task is used to control your robot during the autonomous phase of \*/

/\* a VEX Competition. \*/

/\* \*/

/\* You must modify the code to add your own robot specific commands here. \*/

/\*---------------------------------------------------------------------------\*/

task autonomous()

{

 // ..........................................................................

 // Insert user code here.

 // ..........................................................................

 // Remove this function call once you have "real" code.

 AutonomousCodePlaceholderForTesting();

}

/\*---------------------------------------------------------------------------\*/

/\* \*/

/\* User Control Task \*/

/\* \*/

/\* This task is used to control your robot during the user control phase of \*/

/\* a VEX Competition. \*/

/\* \*/

/\* You must modify the code to add your own robot specific commands here. \*/

/\*---------------------------------------------------------------------------\*/

task usercontrol()

{

 // User control code here, inside the loop

 while (true)

 motor[port2] = vexRT[Ch2];

 motor[port1] = vexRT[Ch2];

 motor[port4] = vexRT[Ch3];

 motor[port5] = vexRT[Ch3];

// if(vexRT[Btn7U] == 1)

// {

 // motor[MobileGoalA] = 40;

 }

 //else if(vexRT[Btn7D] == 1)

 //{

 //motor[MobileGoalA] = -40;

 //}

 //else

 //{

 //motor[MobileGoalA] = 0;

 //}

 //if(vexRT[Btn8U] == 1)

 //{

 //motor[MobileGoalB] = 40;

 //}

 //else if(vexRT[Btn8D] == 1)

 //{

 // motor[MobileGoalB] = -40;

// }

 //else

 //{

 //motor[MobileGoalB] = 0;

 // }

//}