#pragma config(Motor, port2, leftMotor, tmotorServoContinuousRotation, openLoop, reversed)

#pragma config(Motor, port3, rightMotor, tmotorServoContinuousRotation, openLoop)

#pragma config(Motor, port4, armMotor1, tmotorServoContinuousRotation, openLoop)

#pragma config(Motor, port5, armMotor2, tmotorServoContinuousRotation, 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)

#pragma autonomousDuration(20)

#pragma userControlDuration(120)

//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()

{

motor[leftMotor] = 127;

motor[rightMotor] = 127;

wait1Msec(1000);

motor[leftMotor] = 0;

motor[rightMotor] = 0;

}

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

/\* \*/

/\* 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[leftMotor] = vexRT[Ch3];

motor[rightMotor] = vexRT[Ch2];

if(vexRT[Btn6U] == 1)

{

motor[armMotor2] = -63;

}

else if (vexRT[Btn6D] == 1)

{

motor[armMotor2] = 63;

}

else

{

motor[armMotor2] = 0;

}

if(vexRT[Btn5U] == 1)

{

motor[armMotor1] = -63;

}

else if (vexRT[Btn5D] == 1)

{

motor[armMotor1] = 63;

}

else

{

motor[armMotor1] = 0;

}

}

}