#pragma config(Motor, port1, motor, tmotorVex269\_HBridge, 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;

 int lcdScreenMin = 1;

 int lcdScreen = 1;

 int lcdScreenMax = 3;

 bLCDBacklight = true;

 const short leftButton = 1;

 const short centerButton = 2;

 const short rightButton = 4;

 while(vrDisabled == 1)

 {

 if(nLCDButtons == leftButton)

 {

 if(lcdScreenMin == lcdScreen)

 {

 lcdScreen = lcdScreenMax;

 wait1Msec(250);

 }

 else

 {

 lcdScreen--;

 wait1Msec(250);

 }

 }

 if(nLCDButtons == rightButton)

 {

 if(lcdScreenMax == lcdScreen)

 {

 lcdScreen = lcdScreenMin;

 wait1Msec(250);

 }

 else

 {

 lcdScreen++;

 wait1Msec(250);

 }

 }

 else if(nLCDButtons == leftButton)

 {

 if(lcdScreenMax == lcdScreen)

 {

 lcdScreen = lcdScreenMax;

 wait1Msec(250);

 }

 else

 {

 lcdScreen--;

 wait1Msec(250);

 }

 }

 if(lcdScreen == 1 && Program != 1)

 {

 displayLCDCenteredString(0, "program");

 displayLCDCenteredString(1, "1");

 if(nLCDButtons == centerButton)

 {

 Program = lcdScreen;

 displayLCDCenteredString(0, "autonomous has");

 displayLCDCenteredString(1, "been selected");

 wait1Msec(1500);

 }

 }

 else if(lcdScreen == 1 && Program == 1)

 {

 displayLCDCenteredString(0, "program");

 displayLCDCenteredString(1, "[1]");

 }

 else if(lcdScreen == 2 && Program !=2)

 {

 displayLCDCenteredString(0, "program");

 displayLCDCenteredString(1, "2");

 if(nLCDButtons == centerButton)

 {

 Program = lcdScreen;

 displayLCDCenteredString(0, "autonomous has");

 displayLCDCenteredString(1, "been selected");

 wait1Msec(1500);

 }

 }

 else if(lcdScreen == 2 && Program == 2)

 {

 displayLCDCenteredString(0, "program");

 displayLCDCenteredString(1, "[2]");

 }

 else if(lcdScreen == 3 && Program !=3)

 {

 displayLCDCenteredString(0, "program");

 displayLCDCenteredString(1, "3");

 if(nLCDButtons == centerButton)

 {

 Program = lcdScreen;

 displayLCDCenteredString(0, "autonomous has");

 displayLCDCenteredString(1, "been selected");

 wait1Msec(1500);

 }

 }

 else if(lcdScreen == 3 && Program == 3)

 {

 displayLCDCenteredString(0, "program");

 displayLCDCenteredString(1, "[3]");

 }

 }

 // 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()

{

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

 if(Program == 1)

 {

 motor[port1] = 50;

 wait(2);

 motor[port1] = 0;

 }

 else if(Program == 2)

 {

 motor[port1] = 60;

 wait(4);

 motor[port1] = 0;

 }

 else if(Program == 3)

 {

 motor[port1] = 70;

 wait(6);

 motor[port1] = 0;

 }

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

 // 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)

 {

 // This is the main execution loop for the user control program.

 // Each time through the loop your program should update motor + servo

 // values based on feedback from the joysticks.

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

 // Insert user code here. This is where you use the joystick values to

 // update your motors, etc.

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

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

 UserControlCodePlaceholderForTesting();

 }

}