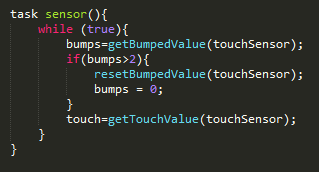
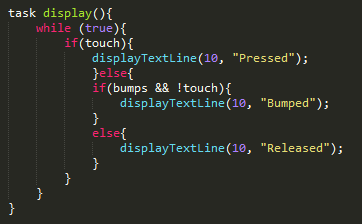
Matt Donnelly

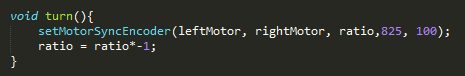
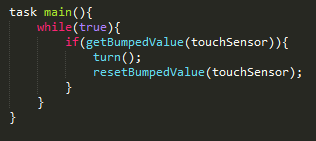
Daniel Bjorklund

Lab 4 report

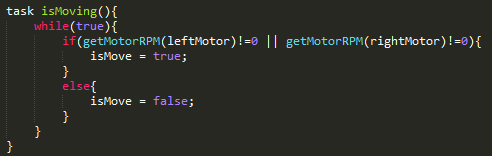
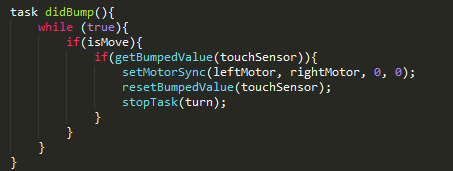
1a. We found this first part to be easy, as we already know how to use and make a display task. The only new part we had to get to know was the touch sensors. We were able to guess (before we did the reading) that the sensor would be using a Boolean to tell us the status of the button. We have the sensor and display running always, both on separate tasks. The sensor sets a variable bumps equal to whatever the value of the touch sensor is. Bumps is reset after the value is greater than 2. We also have a value touch which does the same thing, but that does not reset the value. In the display task we check to see if touch has a value, if so the button is pressed. Else if bump and not touch then we know the button has been bumped. The last resort means the button is released.

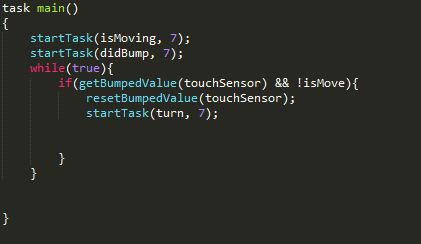
 

1b. We found this question to be pretty simple. We just check the value of the touch sensor. If the value is one (anything but zero), then we call turn and reset the value to zero. Turn turns the vehicle 360 degrees. We have ratio which determines the direction of the next spin, by changing the sign.

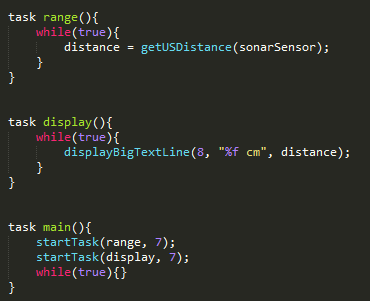
 

1c. We use the same method above, but this time we changed the main to a task as didBump. This task checks to see if the robot is moving, if so it then checks stops the motors and resets the bump value. We also stop the turn task, this is to ensure everything has stopped. We added isMoving to check if the motor is moving or not. This is used in didBump so it knows to stop or not. In the main we then checked to see if it isMoving and if the bump sensor has been hit. If the robot is not moving then it will start the turn, if not then it will not do anything.





2a. This was simple as we had a display task running which took a global variable from range and displayed it on the LCD. We used getUSDistance to figure out how far an object is, an dhtat was assigned to distance. The main task just starts the other tasks and keeps running.



2b. Once again, we have a display task which displays the current distance from the nearest object. The distance is a global variable which is assigned in the range task. The range task just assigns the distance from the sensor. Main task checks the distance, then if it is less than 60 mm away, it will call the new controller task else the power is 0. The controller task checks the distance and the and the error. It then determines if the robot needs to go forward or backward.

