Task 1

task main()

{

motor[motorA]= 50; // Turn motor A on 50%

motor[motorC]= 50; // Turn Motor C on 50%

wait1Msec (3500); // Delay

bFloatDuringInactiveMotorPWM = false; //motors will brake when power is set to "0"

motor[motorA]= 0; // Turn Motor A off

motor[motorC]= 0; // Turn Motor C off

motor[motorA]= 50; // Turn motor A on 50% (Turn around)

wait1Msec (1800); // Delay

motor[motorA]= 0; // Turn Motor A off

wait1Msec (500); // Delay

motor[motorA]= 50; // Turn motor A on 50%

motor[motorC]= 50; // Turn Motor C on 50%

wait1Msec (3500); // Delay

motor[motorA]= 0; // Turn Motor A off

motor[motorC]= 0; // Turn Motor C off

}

Task Main

Turn on Motor A & Motor C

Wait time length (for bot to travel half table length)

Turn off Motor A & Motor C

Wait time length (for bot to not move forward and turn at the same time)

Turn on Motor A

Wait time length (for bot turn 180 degrees)

Turn off Motor A

Wait time length (for bot to not move forward and turn at the same time)

Turn on Motor A & Motor C

Wait time length (for bot to travel half table length)

End Task Main

Task 2

task main()

{

motor[motorA]= 50; // Turn motor A on 50%

motor[motorC]= 50; // Turn Motor C on 50%

wait1Msec (3250); // Delay

bFloatDuringInactiveMotorPWM = false; //motors will brake when power is set to "0"

motor[motorA]= 0; // Turn Motor A off

motor[motorC]= 0; // Turn Motor C off

nMotorEncoder [motorA] = 0;

while (nMotorEncoder [motorA] < 900)

{

motor[motorA]= 50;

displayTextLine (1, "%d", nMotorEncoder[motorA]);

displayTextLine (2, "%d", nMotorEncoder[motorC]);

}

motor[motorA]= 0; // Turn Motor A off

wait1Msec (800); // Delay

wait1Msec (500); // Delay

motor[motorA]= 50; // Turn motor A on 50%

motor[motorC]= 50; // Turn Motor C on 50%

wait1Msec (3250); // Delay

motor[motorA]= 0; // Turn Motor A off

motor[motorC]= 0; // Turn Motor C off

}

Task Main

Turn on Motor A & Motor C

Wait time length (for bot to travel half table length)

Turn off Motor A & Motor C

Wait time length (for bot to not move forward and turn at the same time)

Set Rotational Sensor to 0

While Rotational Sensor < the value needed to turn halfway

Turn on Motor A

Display text on LCD

End While

Wait time length (for bot turn 180 degrees)

Turn off Motor A

Wait time length (for bot to not move forward and turn at the same time)

Turn on Motor A & Motor C

Wait time length (for bot to travel half table length)

End Task Main

Task 3

Top of Form

Bottom of Form

task main()

{

motor[motorA]= 50; // Turn motor A on 50%

motor[motorC]= 50; // Turn Motor C on 50%

wait1Msec (3250); // Delay

bFloatDuringInactiveMotorPWM = false; //motors will brake when power is set to "0"

motor[motorA]= 0; // Turn Motor A off

motor[motorC]= 0; // Turn Motor C off

nMotorEncoder [motorA] = 0;

while (nMotorEncoder [motorA] < 450)

{

motor[motorA]= 50;

displayTextLine (1, "%d", nMotorEncoder[motorA]);

displayTextLine (2, "%d", nMotorEncoder[motorC]);

}

motor[motorA]= 0; // Turn Motor A off

wait1Msec (1300); // Delay

motor[motorA]= 50; // Turn motor A on 50%

motor[motorC]= 50; // Turn Motor C on 50%

wait1Msec (1625); // Delay

motor[motorA]= 0; // Turn Motor A off

motor[motorC]= 0; // Turn Motor C off

wait1Msec (1300); // Delay

nMotorEncoder [motorA] = 0;

while (nMotorEncoder [motorA] < 900)

{

motor[motorA]= 50;

displayTextLine (1, "%d", nMotorEncoder[motorA]);

displayTextLine (2, "%d", nMotorEncoder[motorC]);

}

motor[motorA]= 0; // Turn Motor A off

playSound(soundUpwardTones); //Plays the sound "Upward Tones"

wait1Msec (1300); // Delay

}

Task Main

Turn on motor A & C

Wait time

Turn off motor A & C

Set rotational sensor to zero

While Rotational Sensor < the value needed to turn 90 degrees

Turn on Motor A

Display text on LCD

End while

Turn off motor A

Wait time

Turn on motor A & C

Wait time

While Rotational Sensor < the value needed to turn 90 degrees

Turn on Motor A

Display text on LCD

End while

Turn off motor A

Wait time

Play sound

Wait time

End task main

Task 4

task main()

{

motor[motorA]= 50; // Turn motor A on 50%

motor[motorC]= 50; // Turn Motor C on 50%

wait1Msec (1700); // Delay

bFloatDuringInactiveMotorPWM = false; //motors will brake when power is set to "0"

motor[motorA]= 0; // Turn Motor A off

motor[motorC]= 0; // Turn Motor C off

nMotorEncoder [motorC] = 0;

while (nMotorEncoder [motorC] < 450)

{

motor[motorC]= 50;

displayTextLine (1, "%d", nMotorEncoder[motorA]);

displayTextLine (2, "%d", nMotorEncoder[motorC]);

}

motor[motorC]= 0; // Turn Motor A off

wait1Msec (1500); // Delay

motor[motorA]= 50; // Turn motor A on 50%

motor[motorC]= 50; // Turn Motor C on 50%

wait1Msec (5050); // Delay

motor[motorA]= 0; // Turn Motor A off

motor[motorC]= 0; // Turn Motor C off

nMotorEncoder [motorA] = 0;

while (nMotorEncoder [motorA] < 620)

{

motor[motorA]= 50;

displayTextLine (1, "%d", nMotorEncoder[motorA]);

displayTextLine (2, "%d", nMotorEncoder[motorC]);

}

motor[motorA]= 0; // Turn Motor A off

wait1Msec (1500); // Delay

motor[motorA]= 50; // Turn motor A on 50%

motor[motorC]= 50; // Turn Motor C on 50%

wait1Msec (1200); // Delay

motor[motorA]= 0; // Turn Motor A off

motor[motorC]= 0; // Turn Motor C off

nMotorEncoder [motorA] = 0;

while (nMotorEncoder [motorA] < 620)

{

motor[motorA]= 50;

displayTextLine (1, "%d", nMotorEncoder[motorA]);

displayTextLine (2, "%d", nMotorEncoder[motorC]);

}

motor[motorA]= 0; // Turn Motor A off

wait1Msec (1500); // Delay

motor[motorA]= 50; // Turn motor A on 50%

motor[motorC]= 50; // Turn Motor C on 50%

wait1Msec (5000); // Delay

motor[motorA]= 0; // Turn Motor A off

motor[motorC]= 0; // Turn Motor C off

nMotorEncoder [motorC] = 0;

while (nMotorEncoder [motorC] < 450)

{

motor[motorC]= 50;

displayTextLine (1, "%d", nMotorEncoder[motorA]);

displayTextLine (2, "%d", nMotorEncoder[motorC]);

}

motor[motorC]= 0; // Turn Motor A off

wait1Msec (1500); // Delay

motor[motorA]= 50; // Turn motor A on 50%

motor[motorC]= 50; // Turn Motor C on 50%

wait1Msec (1500); // Delay

motor[motorA]= 0; // Turn Motor A off

motor[motorC]= 0; // Turn Motor C off

}