File: C:\Users\M4rc05\Documents\Vex\Starstruck\2223-G\3-30-2017\Challenge\Challenge.c

```
#pragma config(Sensor, dgtl1,  RightEncoder,    sensorQuadEncoder)
#pragma config(Sensor, dgtl3,  LeftEncoder,     sensorQuadEncoder)
#pragma config(Motor,  port1,           RightMotor,    tmotorVex393 HBridge, openLoop, reversed, encoderPort, dgtl3)
#pragma config(Motor,  port10,          LeftMotor,     tmotorVex393_HBridge, openLoop, encoderPort, dgtl1)
//*!!Code automatically generated by 'ROBOTC' configuration wizard               !!*//

float KP = .25, bias = 0;
static int iterationTime = 1, errorLeft, errorRight, priorErrorLeft, priorErrorRight, outputLeft, outputRight;

static void move(int direction, int maxSpeed, int pulses){
  //direction 1 = forward, 2 = backwards, 3 = left, 4 = right

  errorLeft = 0;
  errorRight = 0;
  priorErrorLeft = 0;
  priorErrorRight = 0;
  outputLeft = 0;
  outputRight = 0;
  SensorValue[LeftEncoder] = 0;
  SensorValue[RightEncoder] = 0;

  while(abs(SensorValue[LeftEncoder])<= pulses || abs(SensorValue[RightEncoder])<=pulses){

    errorLeft = pulses - abs(SensorValue[LeftEncoder]);     errorRight = pulses - abs(SensorValue[RightEncoder]);
    outputLeft = KP*errorLeft + bias;     outputRight = KP*errorRight + bias;
    priorErrorLeft = errorLeft;     priorErrorRight = errorRight;

    if(outputLeft > maxSpeed) outputLeft = (maxSpeed*10)/9; else if (outputLeft < -(maxSpeed)) outputLeft = -((maxSpeed*10)/9);
    if(outputRight > maxSpeed) outputRight = (maxSpeed*10)/9; else if (outputRight < -(maxSpeed)) outputRight = -((maxSpeed*10)/9);

    switch(direction){
    case(1):
      motor[LeftMotor] = outputLeft;
      motor[RightMotor] = outputRight;
      break;
    case(2):
      motor[LeftMotor] = -(outputLeft);
      motor[RightMotor] = -(outputRight);
      break;
    case(3):
      motor[LeftMotor] = -(outputLeft);
```

```
      motor[RightMotor] = outputRight;
      break;
    case(4):
      motor[LeftMotor] = outputLeft;
      motor[RightMotor] = -(outputRight);
      break;
    }
    wait1Msec(iterationTime);
    if (errorLeft<0 || errorRight<0)break;
  }
  motor[LeftMotor] = 0;
  motor[RightMotor] = 0;
  wait1Msec(100);
}

task main(){
  clearDebugStream();
  move(1,100,360);  //Move forwards at speed of 50 for 360 pulses
  writeDebugStreamLine("Finished first step");
  move(4,100,250);  //Rotate 90° right at speed of 50
  writeDebugStreamLine("Finished second step");
  move(3,100,500);  //Rotate 180° left at speed of 50
  writeDebugStreamLine("Finished third step");
  move(2,100,360);  //Move backwards at speed of 50 for 360 pulses
  writeDebugStreamLine("Finished!");
}
```