File: C:\Users\brand\Documents\ME101\ROBOTC\TANGERINE.c

```
/*
  Brandon Choi, Ivan Chen Group 19
*/

void configureAllSensors(){
  SensorType[S1] = sensorEV3_Touch;
  SensorType[S2] = sensorEV3_Ultrasonic;
  SensorType[S3] = sensorEV3_Color;
  wait1Msec(50);
  SensorMode[S3] = modeEV3Color_Color;
  wait1Msec(50);
  SensorType[S4] = sensorEV3_Gyro;
  wait1Msec(50);
  SensorMode[S4] = modeEV3Gyro_Calibration;
  wait1Msec(50);
  SensorMode[S4] = modeEV3Gyro_RateAndAngle;
  wait1Msec(50);

}
void rotateRobot(int angle, int motorPower){
  motorPower = abs(motorPower);
  if (angle > 0){
  motor[motorA] = -motorPower;
  motor[motorD] = motorPower;

  }
  else{
  motor[motorA] = motorPower;
  motor[motorD] = -motorPower;

  }
  resetGyro(S4);
  while (abs(getGyroDegrees(S4)) < abs(angle))
  {}
  motor[motorA] = motor[motorD] = 0;

}

void drive_50(float  distance, int & direction){
  nMotorEncoder[motorA] = 0;
  motor[motorA] = motor[motorD] = 50;
  while(nMotorEncoder[motorA]/180*(2.75*PI) < distance) // convert  angle to cm
  {
    displayString (5, "%.2f", nMotorEncoder[motorA]/180*(2.75*PI));
    if(SensorValue[S3] == (int)colorRed)
    {
      distance = nMotorEncoder[motorA]/180*(2.75*PI);//updates the condition
      displayString (3, "%.2f", distance);
      direction *= -1; //switches CW - CCW
      rotateRobot(180, 25);

      nMotorEncoder[motorA] = 0;
      motor[motorA] = motor[motorD] = 50;
    }

  }
  motor[motorA] = motor[motorD] = 0;
  rotateRobot(90*direction, 25);
}
```

File: C:\Users\brand\Documents\ME101\ROBOTC\TANGERINE.c

```
void display(int group, int line){
  displayString (line, "Group %i BC IC", group);
}

task main()
{
  configureAllSensors();
  display(19,1);
  while(!getButtonPress(buttonEnter))
  {}
  while(getButtonPress(buttonEnter))
  {}
  float distance = 50;
  int direction = 1;
  time1[T1] = 0;
  while (time1[T1] < 1*60000)//while time less than 1min
  {
    drive_50(distance, direction);
    distance = 50;//resets condition

  }

}
//Assumptions
/*
  assumes motor encoder tracks the true travel distance
  assume gyro tracks the true degree of rotation
*/
// Problems
/*
  It cannot check for objects in path

  */
```