```
#pragma config(Motor,  motorA,          rightMotor,    tmotorNXT, PIDControl, er
#pragma config(Motor,  motorC,          leftMotor,     tmotorNXT, PIDControl, er
//*!!Code automatically generated by 'ROBOTC' configuration wizard

#include "joystickdriver.c"

// Define a variable that can be adjusted for setting max power to < 100%.
const bool kMaximumPowerLevel = 100; // Adjust to set max power level to be usec
// For example, let's say your robot arm only needed 50% of max power
// By setting kMaximumPowerLevel = 50, the arm would see smooth control over all
// joystick values and would not be as sensitive if 100 was the max motor power.

////////////////////////////////////////////////////////////////////////////////
//
// scaleJoystick
//
// Function to scale a joystick value using a logarithmic like scale with a deac
// band around the center point, making it easier to control the robot at slow s
//
// Most of the adjustment range is used for fine control over low power settings
//
// The extreme end of the range provides coarse control at high power.
//
// Note the Excel file in our RobotC directory to help see the effects of this
// function graphically on the motor values.
//
////////////////////////////////////////////////////////////////////////////////

int scaleJoystick(int &nJoy1, int nMaxValue = kMaximumPowerLevel)
{
  //
  // This function scales the joystick y1 values to the appropriate range for
  // controlling a NXT motor.
  //
  // Joystick value y1 ranges from -128 to +127.
  // Speed/power settings for NXT motors range from -100 to +100.
  //
  // The physical range of motion of a joystick is quite small and it is sometim
  // hard to control slow speed movements. So we'll apply
  // a "logarithmic" scale to the joystick settings to 'fix' that problem.
  //
  static const int nLogScale[17] =  // This is just an array used to provide the
  {                         // values depending on the joystick input values.
  0, 5, 9, 10,              // If you plot these values over joystick values of
  12, 15, 18, 24,           // you'll see the log scale this generates.
  30, 36, 43, 50,           // See Excel spreadsheet in the RobotC directory.
  60, 72, 85, 100,
  100
  };
  int nScaled;

  nScaled = nJoy1;  // Assign the joystick input value to the variable nScaled.

  nScaled /= 8;  // We divide the joystick input values by 8 which results in 16
                 // These will be the 17 values (16 plus one for a zero value) s
  if (nScaled >= 0)  // If the joystick value is positive, then we apply positiv
  nScaled = nLogScale[nScaled];  // from the array above to the joystick input v
  else  // If the joystick values are negative, then we set the motor values to
  nScaled = - nLogScale[ - nScaled];
```

```
  nScaled *= nMaxValue;  // This is where you can apply scaling to the final mot
  nScaled /= 100;        // as described above.  This looks useless when you war
 return nScaled;         // but makes sense when you want say 50%.
}

void Arcade(int x, int y) // Uses joystick arcade style controller
{
 int powY;             // y is the Power or Speed
 int powRightMotor;    // x is the turn power
 int powLeftMotor;

 // convert joystick -128 to 127 range using the array numbers to produce a log
 powY = scaleJoystick(y); // joystick y axis gives the power level

 if (x < 0) // if x negative, turning left; otherwise, turning right
 {
 powLeftMotor = (powY * (128 + (2 * x))/128); // left motor reduced for left tur
 powRightMotor = powY; // right motor not changed
 }
 else
 {
 powRightMotor = (powY * (128 - (2 * x))/128); // right motor reduced for right
 powLeftMotor = powY; // left motor not changed
 }

 motor[leftMotor] = powLeftMotor;
 motor[rightMotor] = powRightMotor;
}

// main
//
// Drive control example main line code. Uses the first game controller to drive
// a simple two motor NXT bot.

task main()
{
 while (true)
 {
 getJoystickSettings(joystick); // Get the values of the joysticks x and y.

 Arcade(joystick.joy1_x2, joystick.joy1_y2);  // Uses the right joystick for Arc
                                              // Change joystick.joy1 to joystic
 wait10Msec(1);                               // if you want to use the left joy
 }
}
```