# Setting up Nav and Control configuration files

Now that all the microcontroller configuration data has been moved to the laptop, this is getting a bit complicated. So here is some clarification. (someday this will be more comprehensive)

**NavAndControl Config file (.ini)**

This window sets some basic parameters for the windows used by Nav and control. Generally, you won't have to do anything with these. But, if you do, most of the parameters are the standard stuff used when you create a window. Width and height are in pixels. X/Y are the upper left coordinate of the window and the Title goes on the top of the main window.

The animation On parameter determines whether the "Rocky" simplified robot face animation is used or not.

**MicroConfig configuration file (.ini)**

This file determines how much of the microcontroller's I/O is handled.

[Config] sets the number of sonars used. The remaining ports will be for RC control. It also sets the input or output status of each discrete I/O bit.

[Analog] is provisions for some filtering etc of analog signal. Not used yet.

[RCservo] contains calibrations for each RC servo. The nominal values of 1000 and 0 represent the gain and displacement of the servo. If your servo rotates farther than commanded , reduce the 1000 value. If the servo is not centered correctly, change the zero value.

**Microcontroller configuration file (.ini)**

This file controls the drive motor configuration.

The [odometry] numbers provide a numerator and a denominator for a fudge factor which converts encoder counts to millimeters. So, you vary these number to make your robot go exactly the right distance.

The [Motors] section has the gains to set up the motor drivers.

MotorType determines whether the board outputs H-bridge or RC commands.

RCoffsetL/R allows the RC command output to be biased a little higher or lower to ensure that a motor controller actually commands zero when told to do so.

PWMmax restricts the maximum signal which can be commanded. The value can be between 0 and 100 for H-bridge operation. It is normally 100, but can be reduced to limit motor torque. This same number can be used to reduce motor torque with RC motor controllers but can also be increased above 100 if necessary to get full power out of a motor controller.

PWMbase L/R is a value in %duty cycle which can be used to make motor operation more linear; For example, if you have a motor that takes 30% duty cycle before it even starts moving (cheap motor!), you can make this value 30 to make the drive system more accurate. Normally this number will be zero.

KvL/R, KaccL/R, and KdistL/R are gains for the PID control equations. See the Drive Motor Integration instructions to set these.

**Animation configuration file (.ini)**

The Emotion matrix determines how much each facial feature moves for specific emotion data sent by the lisp software.

The Feature Limits specify the minimum, neutral and maximum values for each facial feature.

See the Animation.cpp source code to figure out how to use these.

**Sensors configuration file (.ini)**

This file has calibration factors for the various types of sensors.  See the GP2D12 and SRF04 and Battery.cpp files to see how these are used.  In general, calibration for each sensor has a scale factor (usually called calibration) and an offset number.  Usually, the basic conversion from input signal to engineering units (e.g. millimeters) is done in the sensor class.  Hence, a calibration number of 1000 is used if the sensor agrees with the calibration.  If the sensor has an error which is different at long range than at short range, the calibration number can be change to improve the reading.  If the reading is off by the same amount at all ranges, the offset number can be changed.