Using the PSoC nodes

To start up the PSoC nodes first source the setup script from the workspace that the PSoC nodes are a member of. Then launch the PSoC nodes. In Ubuntu this can be done with the following sequence:

open a new terminal

roscore

open a new terminal or terminal tab

source <workspace path>/devel/setup.sh

source <workspace path>/devel/setup.bash

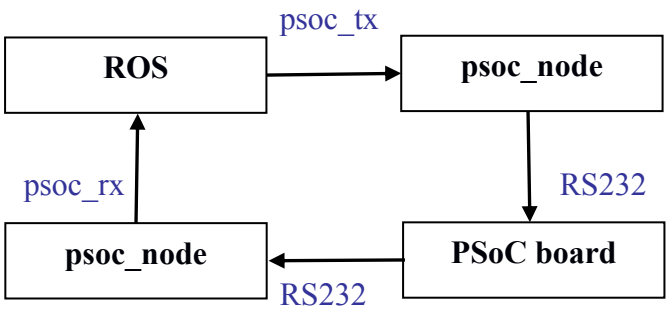
roslaunch serial_test psoc_launch.launch

The node will automatically detect the port each PSoC is on and handle all exceptions and report problems to the terminal. In the event in an unexpected failure of the node, the node will relaunch and attempt to reconnect to the PSoC.

An xterm window can be used to display the data. If this is needed simply uncomment the launch file line for the listener node.

<workspace path>/src/serial_test/launch/psoc_launch.launch

Now that the node is running you can send data to the PSoC using the psoc_node. This node works by taking an input string off of the psoc_tx topic and serializing it to be written to the PSoC.



ROS programs will publish String data to the psoc_node using the psoc_tx topic. The psoc_node will then send this string data to the PSoC board using RS232 serial data. This is handled automatically so you do not need to do anything special, just send an appropriate data string.

When the PSoC board receives a valid data string it will respond with return data. This will go over RS232 serial to the psoc_node. To read this data you need to subscribe to the psoc_rx topic. The data will be given back as a string so you will need to do a conversion depending on what data type the string represents. See the table below for this info.

Command	Description	Format	Return Value
L or l	Steer left	"L <floating point angle>\r"	none
R or r	Steer right	"R <floating point angle>\r"	none
F or f	Move forward	"F <integer speed>\r"	none
B or b	Move backward	"B <integer speed>\r"	none
S or s	Get steer enc. count	"S\r"	floating point angle
D or d	Get drive enc. count	"D\r"	floating point speed
P or p	Get steer pot position	"P\r"	int steer shaft pos.
A or a	Get actuator position	"A\r"	int actuator pos.
H or h	Halt	"H\r"	none

*NOTE: All return values will actually be in the form of string. These will need to be converted to their noted data types. If using C++ this is simple by using atof (ascii to float), atoi (ascii to integer) and similar functions from the <stdlib.h>.