

2003/07/19



-PINO

GUI Manual (sample)

Ver. 1.01

PINO GUI

PINO GUI	2
1 INTRODUCTION	3
2 RUNNING THE SOFTWARE	3
3 SYSTEMS TAB	5
3.1 LOAD/SAVE CONFIG	5
3.2 CAN QUERY TIMEOUT GET/SET	5
3.3 SESSION LOG.....	5
3.4 LOAD/SAVE/GO POSE	6
4 MOTOR TAB	6
4.1 ACTIVE MOTOR.....	7
4.2 POWERSTATE	7
4.3 POSITION	7
4.4 HOME (CALIBRATION)	7
4.5 POSITION LIMITS	8
4.6 PWM LIMIT	8
4.7 PID PARAMETERS	8
5 SENSORS TAB	9
5.1 READ SENSOR VALUES	9
5.2 CALIBRATE SENSORS	9
6 RUN MOTION TAB	11
6.1 LOAD MOTIONS	12
6.2 RUN MOTIONS.....	12
6.3 SENSOR SERVO	14
7 CONFIG-FILES (.CFG)	16
7.1 SYNTAX	16
8 MOTION-FILES (.MTN)	17
8.1 SYNTAX	17

1 Introduction

This is a brief introduction to the functionality of the GUI to Pino Version 2. The intention with the GUI is for the user to be able to quickly get started working with Pino and to provide functional sample code for how the pino2core.dll (API) can be used.

Most things that can be done in the GUI are more or less direct calls to the API. Thus to understand the more detailed parts of the GUI and the settings, refer to the API Reference.

The software is written in C# using Microsoft Visual IDE and .NET framework. The functionality provided is:

- Load/Save configuration of Pino (calibration of joints, motor-control settings etc.)
- Set/Get the CAN Query timeout
- Load/Save and Go to Pose
- Set all or individual motors to power-state On, Brake or Free.
- Set all or individual motors to current position or home position.
- Set/Get motor-control configuration for each motor
- Set home position for each joint (calibration)
- Set/Get max/min positions of joints (limits)
- Set/Get PWM-limit of the motor (motor power)
- Set individual joints to a custom position
- Read the Gyro, Acceleration and Foot force sensors on Pino
- Calibrate the Gyro and Acceleration sensors
- Load and run pre-made motion-files. (.mtn files)
- Use feedback from the gyro-sensor to stabilize the motions.

The GUI consists of four different tab-pages to the right together with a picture of Pino's body to the left. The four tab-pages are System, Motor, Sensors and Run Motion.

2 Running the software

To run the GUI, double-click the MotorControlUI.exe file (or enter the name at the command prompt). It is also required that the pino2core.dll is present, preferably in the same directory as MotorControlUI.exe, or if preferred in the system path.

If the .NET framework is not installed, you need to download and install that from Microsoft's webpage: <http://msdn.microsoft.com/netframework/downloads/howtoget.aspx>

Furthermore to being able to connect to Pino via USB, the Pino USB driver must have been installed.

After double-clicking MotorControlUI.exe, a window will be opened that asks for which (virtual) com-port that Pino is connected to. As an option it is possible start the application without Pino connected, by checking the "without Pino" checkbox and choosing arbitrary com-port.

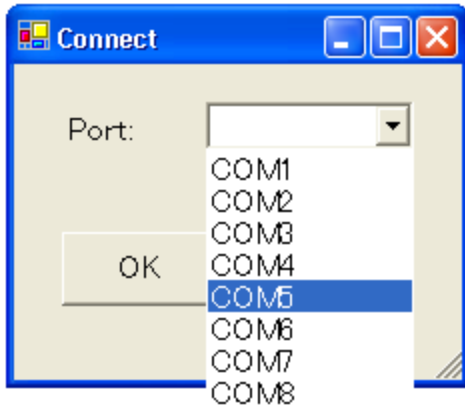


Figure 1. First step when starting the Pino GUI. Here COM5 is chosen.

After a successful connection, the main PINO2 control window should appear, see Figure 2.

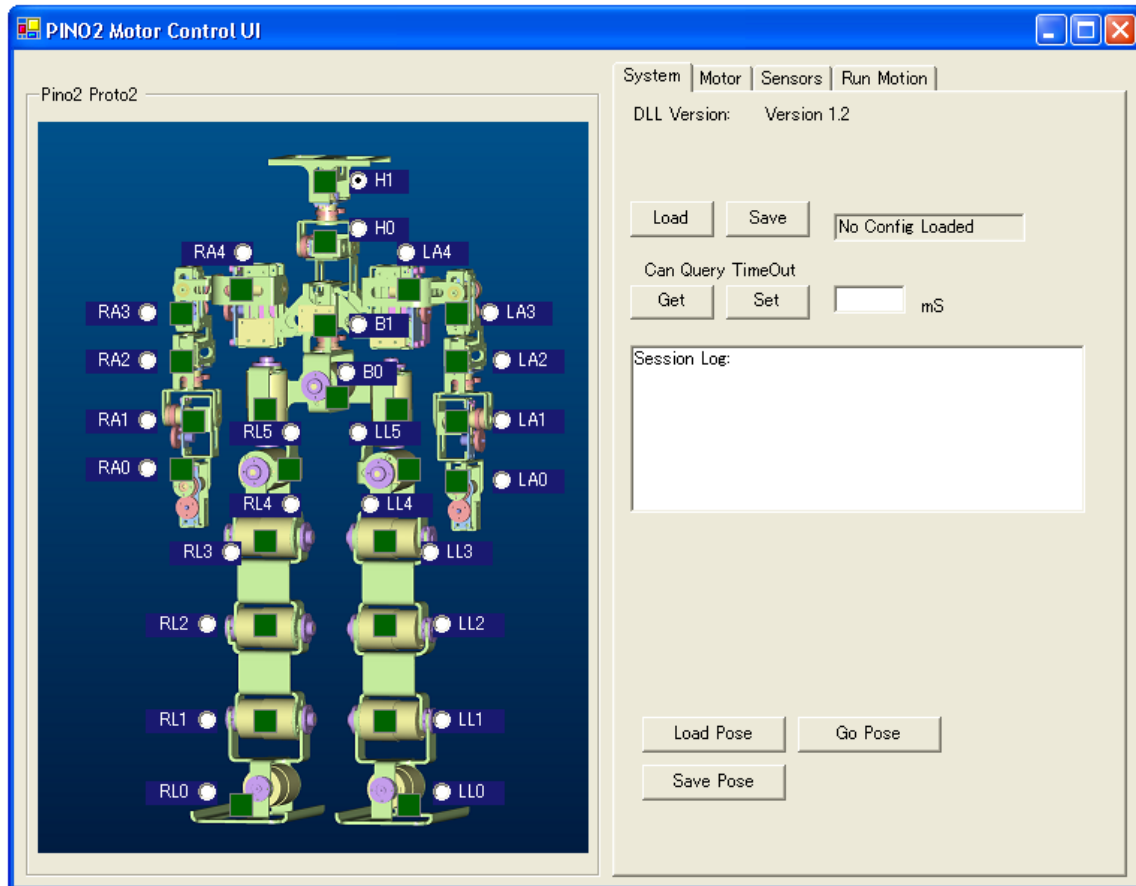


Figure 2. Main Pino2 control window showed after starting the application.

The tab (right part of the window) showed directly when starting is the “System”-tab. Worth noting here is the DLL-version, currently “Version 1.2”. If no DLL could be loaded, it says “DLL Not Loaded”. The most probable reason for that to happen is if the pino2core.dll was not present in the directory of the MotorControlUI.exe (or in the system path).

3 Systems tab

3.1 Load/Save Config

Below the DLL version label are buttons for Loading and Saving a configuration of Pino. Before continuing using Pino a suitable configuration file should always be loaded.

The configuration file (with suffix .cfg) currently stores home-position (default position of a joint), pwm-limit, max/min-position limit and PIDwindow setting for each joint. As well as zero-offset for the X, Y, Z gyro and set parameters for the Sensor Servo. For further details see chapter 7, Config-files (.cfg).

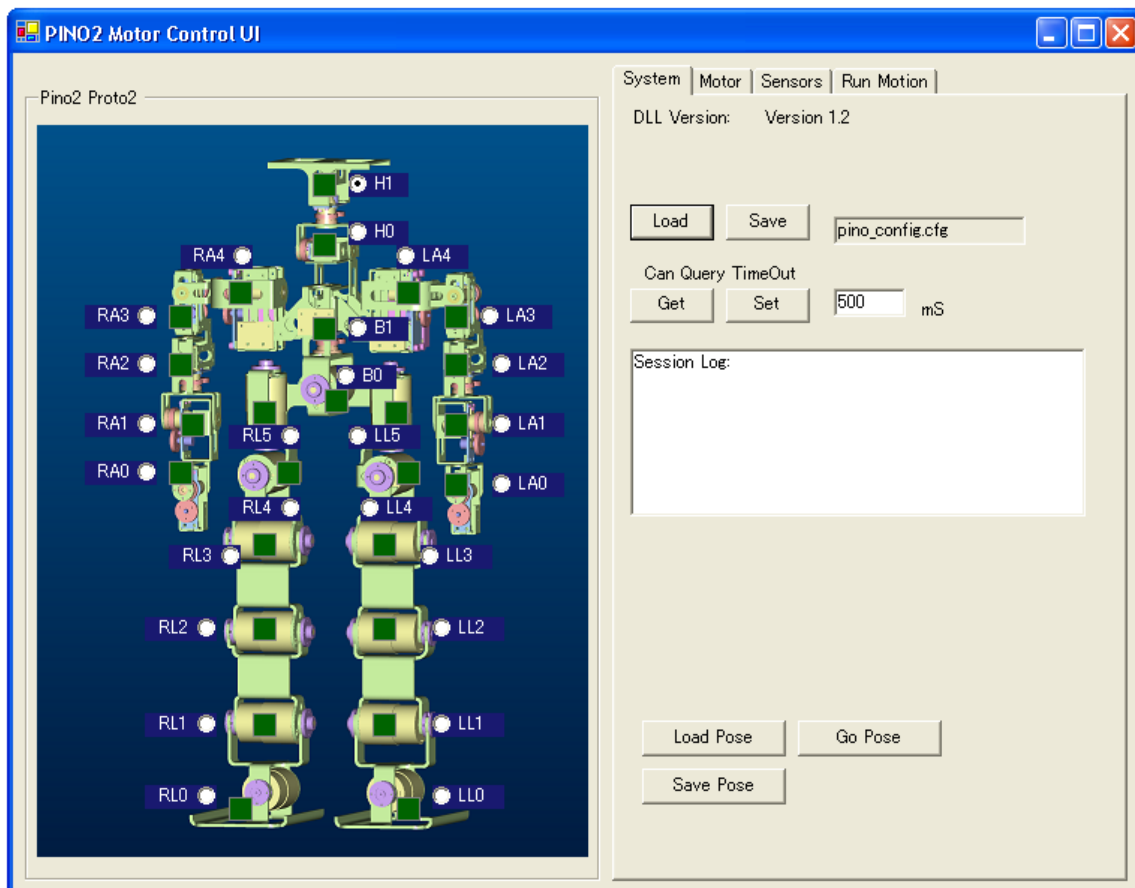


Figure 3. Configuration file named pino_config.cfg loaded.

3.2 CAN Query TimeOut Get/Set

The Can Query timeout is the number of milliseconds the application should wait for a reply if a query for a value (for example sensor reading, joint position) is issued. If no reply is received after this time, an error will be reported. The default setting is 500mS.

3.3 Session Log

In the Session Log, error and warning messages are printed during a session.