Overview

Whisker is installed under Windows. You run the Whisker server (which talks to the hardware) and Whisker clients (which implement behavioural tasks), together with other programs in Windows.

Example of pressing Alt-Tab to switch windows while running Whisker
Hardware

Whisker supports a wide range of hardware, including digital I/O devices from a range of manufacturers...

... touchscreens via the TouchBase UPDD drivers and any serial port supported by Windows...

... any sound card supported by Windows (also allowing the splitting of a single stereo device into two mono sound devices)...

![Configure audio devices](WhiskerServer_ConfigureAudioDevices.bmp)

![Audio Device Summary View](WhiskerServer_AudioDeviceSummaryView.bmp)
... any video card supported by Windows, including multimonitor cards...
... and serial ports used as digital I/O devices.

<table>
<thead>
<tr>
<th>Pin</th>
<th>Signal</th>
<th>Pin</th>
<th>Signal</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Data Carrier Detect</td>
<td>6</td>
<td>Data Set Ready</td>
</tr>
<tr>
<td>2</td>
<td>Received Data</td>
<td>7</td>
<td>Request to Send</td>
</tr>
<tr>
<td>3</td>
<td>Transmitted Data</td>
<td>8</td>
<td>Clear to Send</td>
</tr>
<tr>
<td>4</td>
<td>Data Terminal Ready</td>
<td>9</td>
<td>Ring Indicate</td>
</tr>
<tr>
<td>5</td>
<td>Signal Ground</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Whisker uses the Touch-Base UPDD driver to communicate with touchscreens from any manufacturer supported by UPDD.

Whisker uses the Windows multimonitor facility to drive multiple monitors (with touchscreen attached, if desired).
Whisker allows you to manipulate the on/off state of digital input/output lines directly to test your equipment, and to watch the state of all lines (or all lines being used by a particular task) as tasks are running.

Whisker monitors its own performance, taking care to aim for 1 kHz hardware polling (1 ms temporal resolution) while never consuming more than 50% of CPU time.

Worst inter-poll interval so far [ms]: 2
This display is scheduled to be updated every 1000 ms
Worst inter-poll interval since last update [ms]: 2
Since last update, have had 1004 polls and 3 yields
Of those polls, 100.0% were <10ms, 0.0% were 11-20 ms, 0.0% were >20ms
Longest poll since last update took 96 microseconds
On the high-performance CPU timer, last poll took 186 ticks and last interpoll took 3397 ticks
High-performance CPU timer is running at 3579545 Hz
Server process priority: Real-Time
Clients

Whisker uses a client–server architecture.

Whisker tasks typically write their data direct to an ODBC-compatible database.

Provides data.
Knows how to talk to ODBC.

Provides an interface so that any application can talk to any database, as long as they both "speak" ODBC.

Knows how to handle a specific type of database (e.g., MS Access 97, MS Access 2000, MySQL, Oracle).

Where the data ends up living.
Contains tables with columns (fields) and rows (records).

Application
- e.g. five-choice serial reaction time task
  Communicate using a Data Source Name (DSN) to decide which database to use, e.g. "FiveChoiceAmphetamine".

Open Database Connectivity (ODBC) system
On Windows computers, this is part of Windows
- DSNs are registered with ODBC. For example, ODBC may have been told that the DSN "FiveChoiceAmphetamine" refers to the database file c:\Experiments\Amphetamine\Experiment.mdb, and is accessed by talking to the Microsoft Access 97 ODBC driver.

ODBC driver and database engine
- e.g. Microsoft Access 97 with Microsoft Access 97 ODBC driver

Database file
- e.g. C:\Experiments\Amphetamine\Experiment.mdb

Principles_of_ODBC.BMP
The Whisker Database Manager simplifies the use of ODBC-compatible databases (such as Microsoft Access).

Stimuli can be drawn using any of the Windows drawing primitives, through a simple interface.
The Whisker Visual Basic Software Development Kit makes the process of developing Whisker tasks easy.
Whisker uses a high-speed text-based network communication system. A test client allows you to explore what happens behind the scenes.

Clients can assign names to individual input/output (I/O) lines.

Communication between the client and the server can be monitored for debugging a new client.

For visual displays, clients can manage several documents, each showing different content, and choose which
document to show to the subject.

The server can be used to monitor what any subject is seeing.

The server allows many clients (tasks) to run simultaneously, and allows you to keep an eye on all of them.

Clients can display things on a screen, respond to touchscreen, mouse, and keyboard events, play sounds, turn devices on and off, and create timers.

Whisker provides advanced debugging features, including the facility of ‘fake’ devices (digital I/O, audio devices, etc.), allowing you to write tasks and test them on computers that do not have digital I/O installed.
Existing tasks

Tasks written for Whisker to date include some of the most popular in their class worldwide, including MonkeyCantab...
... second-order schedules of reinforcement...
... the Seeking–Taking Task...
... simple schedules of reinforcement...

... delayed reinforcement choice tasks...
... the Five-Choice Serial Reaction Time Task...

... and many more.