
Buzz - robotics application development system for Windows 95/Windows NT

Article Type: Mini features From: Industrial Robot: An International Journal, Volume 26, Issue 1.

Keywords: Robotics, Robots, Sankyo

Buzz, a new application development system (ADS) software package from Sankyo Seiki Manufacturing Co. Ltd, is a highly advanced, state-of-the-art ADS, designed specifically to work with the Sankyo SC3000 Series Robot Controllers.

What is Buzz?

Buzz is a comprehensive, full function, editor, debugger and communication facility that runs on the Windows 95/Windows NT platform. Major functions include:

- **Creating/editing robot application source files.**
- **Compiling/building source files to executable object files, called tasks.**
- **Full communications support between PC and robot controller.**
- **Debugging robot controller tasks in real time.**
- **Full function file manager.**

Why is Buzz important to robot controllers?

Robot controllers require more than just an editor, more than just a debugger, and more than just a communication facility. Each Sankyo robot application can have up to 13 executing tasks, and each task can be constructed of up to eight source files, and makes for a total of 104 source files. Buzz is a true ADS package, because Buzz combines everything that is required for robot controller "application development" into one simple, state-of-the-art software package.

How does Buzz help create robot applications?

Creating and editing robot application source files with Buzz has the full support of a multi-functional editor. To start with, the programmer can select screen colors for different source file functions. Programming functions, data types, comments, and all other text can be assigned different colors, providing an easy view to the various parts of your programming source files (see Figure 1).

The Buzz editor contains the normal editing functions employing *Undo/Redo*, and *Copy, Cut and Paste*. Several functions are available to help you go immediately anywhere in your source file. These functions are *Find, Find Next, Goto Line* and *Bookmark*. A *Replace* feature is also available with the *Find* and *Find Next* functions. The entering of repetitive text is made user friendly with a *Template* feature. Simply type in your text, and save it with a distinct *Template* number. Now you can recall your repetitive text and use the *Insert, Display* or *Modify Templates* feature. If you have a need to make any part of your source file, or all of your source file, *Upper Case* or *Lower Case*, simply highlight and click.

Buzz can also perform other functions unique to programming editors. Setting your source file to *Read Only, Matching Braces* (parentheses or brackets), and *Printing* your source and teach data files as well. You can have up to 13 executable files called tasks, assigned to a single robot application, and Buzz can compile or "build" all of your source files with one simple click of the mouse.

Once you have completed your source files, you can compile or "build" those tasks with Buzz. Then, you can easily download them to the controller for the debug phase. No need to call other programs, Buzz does it all!

How does the buzz debugger work?

The Buzz debugger works in real time mode with the robot controller. Dynamic setting of the robot arm speed is available at anytime in debug mode. To start your debug session, simply click the *GO* icon, and your tasks are compiled and downloaded, and the PC and robot controller are set to the debug mode. If you do not need to compile or build your tasks, you can just download your tasks with the Buzz file manager.

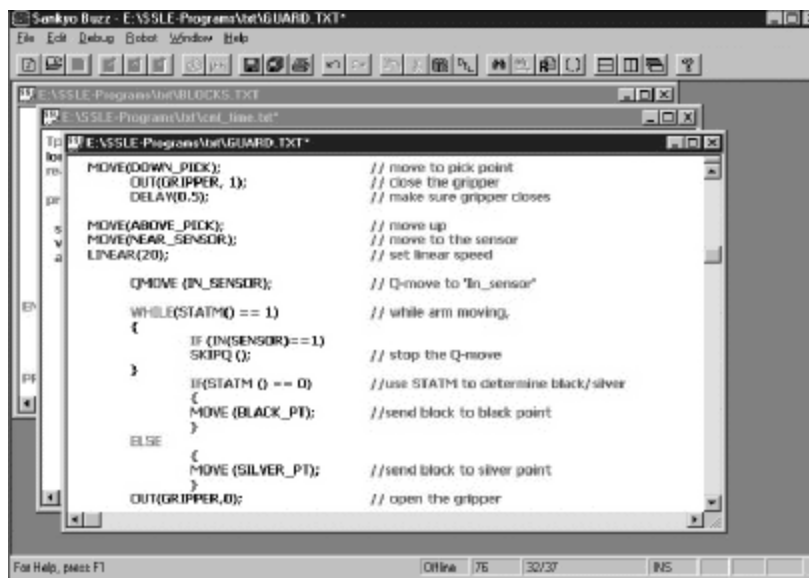
Several debugging tools are available with Buzz. The contents of variables can be observed at any time, and if necessary, changed dynamically. Breakpoints can be set on and off, and cleared. System input/output operations allow task to task control and can be enabled or disabled. And, of course, you can start, single step, stop, hold or resume any of your tasks under debug. Finally, most errors can be reset dynamically.

Need to manage files on your robot controller? Buzz does that too!

For more information contact Joanne Pransky, Sankyo Robotics, 1001-D Broken Sound

Parkway NW, Boca Raton,
Florida 33487, USA. Tel: +1
561 998 9775; Fax: +1 561 998
9778; E-mail:

joanne@sankyo.com



```

Sankyo Buzz - E:\SSLE-Program\test\GUARD.TXT*
File Edit Debug Robot Window Help
E:\SSLE-Program\test\GUARD.TXT
E:\SSLE-Program\test\GUARD.TXT
E:\SSLE-Program\test\GUARD.TXT
// move to pick point
MOVE(DOWN_PICK);
OUT(GRIPPER, 1); // Close the gripper
DELAY(0.5); // make sure gripper closes
// move up
MOVE(ABOVE_PICK);
MOVE(NEAR_SENSOR); // move to the sensor
LINEAR(20); // set linear speed
QMOVE (IN_SENSOR); // Q-move to 'In_sensor'
WHILE(STATM() == 1) // while arm moving,
{
  IF (IN(SENSOR)==1) // stop the Q-move
  SKIPQ Q;
  IF(STATM() == 0) //use STATM to determine black/silver
  {
    MOVE (BLACK_PT); //send black to black point
  }
  ELSE
  {
    MOVE (SILVER_PT); //send black to silver point
  }
}
OUT(GRIPPER,0); // open the gripper

```

Figure 1 Example source file window
