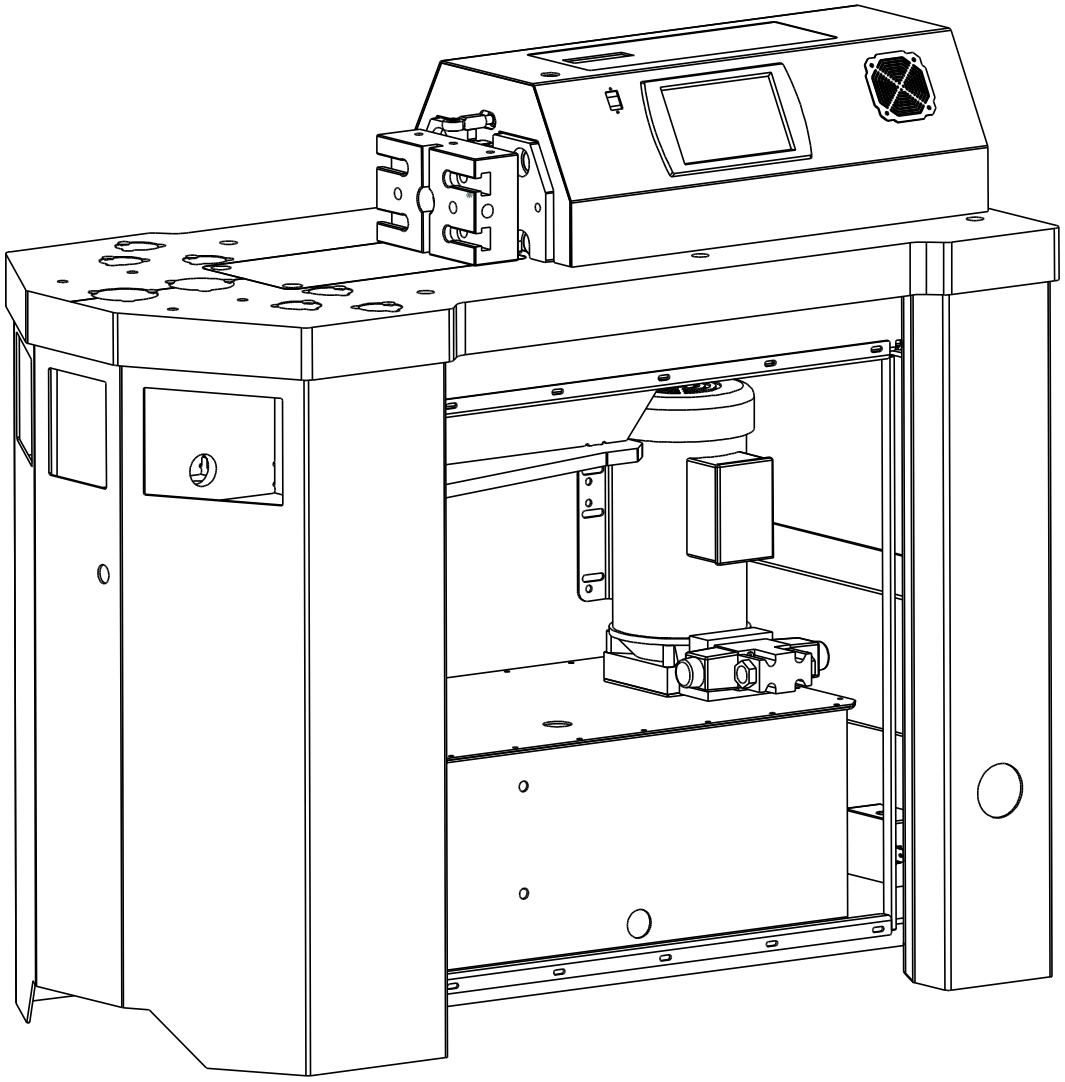
HP-100 HORIZONTAL PRESS

Quick Start Guide



JD Squared Inc.

2244 Eddie Williams Rd. Johnson City, TN 37601
(423) 979-0309, FAX (423) 979-2426
http://www.jd2.com

Introduction

This guide is intended to quickly acclimate you with your HP-100 press. New features include:

- Recalibration support for the touchscreen
- Dynamic choice of operator pedal input modes
- A programming language
- Programmable text and audible operator hints

Please read this booklet thoroughly before beginning operation of your machine.

The HP-100 Press

The HP-100 Press is an automatic piece of machinery and can be very dangerous. It is the user's responsibility to provide all required guarding and proper training to any operators.

Built-in Safety Features

The HP-100 includes many standard safety features. The most important are emergency stop switches, ram return crush-plate detection, and the ram force retract switch. Any emergency stop will immediately halt all motion of the machine and remain halted until all alarms are cleared. The ram force retract switch is located on the lower front of the machine. It allows an operator to use their legs to force the ram to retract in the event their hands are unable. When this switch is pressed, the ram retracts indefinitely unless the emergency stop or a return crush-plate is depressed. The ram return crush-plates halt the ram movement in the event something is behind the ram while it is trying to retract.

For Machines without Proportional Valves

If you do not have the optional proportional valve, then the ram operates in a bang-bang mode similar to a limit switch disengaging the extend or retract signal. Although the press is built with the best components we can purchase, mother nature has made sure the delay times in the physical hardware and the inertia of the moving ram cannot be stopped immediately. As a side effect of this inertia, there is a minimum amount the ram can move which is typically about 0.125” (3.2mm). This does not mean the resolution of the machine is limited to that minimum move value, rather, it simply means the ram must be at least that far away from its target to stop correctly.

To combat this overshoot problem for normal motion, the HP-100 uses the concept of Ram Correction. This value is the amount added to the ram target at which it will trip the “limit switch”. By setting the correction properly, the ram will repeatedly come to rest at the target location. The problem is this value will be different for different materials, thicknesses, and even widths; so the best method of determining it is through trial and error, and recording known correction factors for your press

someplace safe (or even in an automatic program). An example of finding the proper correction value is given in the manual operation section.

The Control Interface

The control interface of the HP-100 is divided into multiple tab pages:

- Manual
- Automatic
- Program Functions
- Settings

Each tab is discussed in the following sections.

Manual Operation Tab

The manual operation tab is the simplest method in which to move the ram. It simply accepts a ram target (the point to which the ram will extend), a ram return point, and a ram correction value. If you alter any value on the page, you must press the ***Set*** button for them to take effect.

Depending on the pedal mode selected, the ram will behave differently. In single pedal mode, pressing the pedal causes the ram to extend to the target point, and releasing causes the ram to retract to the return point. In dual pedal mode, the extend pedal causes to the ram to extend to the target point, and the retract pedal causes the ram to retract to the return point.

For machines without proportional valves

When you are fine adjusting the ram extend point, for either a sharper bend or a shallower bend, do not use the ram correction box. Any bend adjustment is actually a target change. The ram correction box is solely to correct for the amount the ram missed the desired target. To find the proper ram correction value, simply set a target, for instance, 3.0", and zero the ram correction. Then, extend the ram until the machine halts, and note the actual position of the ram. For this example, say the ram moved to 3.052". The proper value to put in the ram correction box is

(DESIRED - ACTUAL) = (3.0 - 3.052) = -0.052

It is important to note that the machine must be operating on the material you intend to use, as this value will be slightly different under varying loads. Once you have given the press this initial guess, simply select the ***Auto Correct Ram*** checkbox and it will perform any necessary correction for you. If you do not put in an initial value, the correction will start from 0, and possibly waste material until the correction converges to the proper value.

Automatic Operation Tab

*Example programs and the proper way to construct them is covered later in this guide, in the **Writing an Automatic Program** section.*

The automatic operation tab is where you will edit and run the currently selected program. It contains the program text box, the current step count, and the three control buttons. To edit the program content, you must enter edit mode, by pressing the ***Edit*** button. To run the program, you must enter run mode by pressing the ***Run*** button. When you press the run button, the HP-100 controller analyzes your program for any errors, and subsequently throws alarms if any problems are found. If there are no errors, the program is ready to execute. To start, simply press the ***Start*** button.

Restarting the program at a specific place

The HP-100 controller fully supports restarting the program from any command block, at any step count in that block. To restart at the beginning of the block, simply touch the desired restart point in **Run** mode, and press the ***Start*** button. The program will advance to the selected commanded block properly setting the control parameters such as modal Ram Return and Ram Correction. Now, to skip ahead in step count, simply change the step count to the desired value and operate the program normally using the pedal(s).

How pedals advance the program

Automatic program execution differs only slightly dependent on the selected pedal mode. In dual pedal mode, the extend pedal controls program actions, and the retract pedal will always force the ram to the machine home position. In single pedal mode, the pedal controls the program actions.

To execute a command block, press and hold the program action pedal until the motion or command is completed, then release. The program will automatically increment to the next command block, or step count if the command is to be executed multiple times.

Program Operations Tab

This tab is used to create, select, or modify the program the HP-100 will execute. It displays the name and description of the program selected, as well as the directory it is stored in. Detailed description of each button follows

New
The new button creates a new program. To select the directory to create it in, press the ***Browse*** button. Give the program a name and a description, then press ***OK***. The newly created program will become the selected program.

Load
Opens a browse dialog box. Select the desired program file (must end with *.txt) .

Save As
Choose the directory to save the program in, give it a new name, and optionally a description. It is not allowed to save a program with a name that is already taken on the hard disk, (e.g. no overwriting).

Delete
Opens a browse dialog box. Select the desired program file (must end with ***.txt***) . If the program is in a directory with the same name as the program, the entire directory is deleted.

Copy Sound
Opens a browse dialog box. Select the desired sound file to copy from the hard disk or USB key and press ***OK***. A second browse dialog is opened. Select the desired destination directory and press ***OK***. The sound will be copied in to the new directory.

Settings Tab

The settings tab exposes machine parameters that are adjustable by the user. They are persistent, in the manner that as soon as they are changed the press saves them to be reused for all future machine operation. Each setting is detailed below.

Units
Select the unit system machine measurements are made in.

Pedal Type
Select the type of pedal connected to the machine. Pedal functionality is described in the ***Manual Operation*** and ***Automatic Operation*** sections of this guide.

Machine has proportional valve
Check this box if your press is equipped with the optional proportional valve.

User text remains past assignment block
Check this box if you wish the user message written by the automatic program ***TEXT*** command to remain in the message box past the command block in which it is assigned. If you unselect this box, the operator is required to cycle the program action pedal to advance the program.

Volume
Adjust the on board speaker volume.

Diagnostics
This button launches the diagnostic window for the machine.

ReHome

Forces the machine to reenter homing mode.

Force Quit

Shuts down the press controller. You should never press this button unless instructed to do so by technical support.

Update

This button launches the update window.

Calibrate

This button launches the touchscreen calibration software. To calibrate the touchscreen follow the steps below.

Step 1 - Press the ***Calibrate*** button on the ***Settings*** tab. A window will pop up with two small tabs on the top.

Step 2 - Select the ***Calibration tab.*** Press the ***Recalibrate*** button. It will launch the recalibration program.

Step 3 - A crosshair will appear in the center of the screen. Accurately press the center of the target, and repeat as it moves around the screen. When the crosshair disappears, press the center of the screen to exit the calibration program.

Step 4 - Press the small ***OK*** button on the popup window to close the calibration tab and return to the HP-100 control program.

Regular calibration of the touchscreen will ensure your finger or dowel presses are correctly interpreted by the user interface.

Writing an Automatic Program

The automatic programming language in the HP-100 is designed to be as powerful as reasonably possible, while still maintaining a very low training requirement for use.

The code in Listing 1 will be used to demonstrate the key components in a program.

```
PROG1 (A SAMPLE PROGRAM)

(THIS IS A COMMENT)

RC=-0.050 (SET A RAM CORRECTION HINT)
RC=AUTO (LET THE RAM AUTO CORRECT FROM NOW ON)

RT=5.0 RRM=4.8 (EXTEND THE RAM AND RETRACT)

(PRINT A MESSAGE TO THE OPERATOR)
TEXT="DO SOMETHING IMPORTANT"

RT=5.1 (EXTEND AND RETRACT THE RAM TO THE RRM)

(THE FOLLOWING BLOCK WILL BE REPEATED 5 TIMES)
(FOR A TOTAL OF 6 RAM EXTEND AND RETRACTS)
RT=5.125 RR=4.5 REPEAT=5 (OVERRIDES RRM)

RT=5.050 (EXTEND AND RETRACT THE RAM TO THE RRM)

RRM=OFF (TURN OFF THE RRM POINT)

RT=5.25 (EXTENDS THE RAM BUT DOES NOT  RETRACT)
RR=1.0 (RETRACTS THE RAM)

(THE NEXT LINE REWINDS THE PROGRAM AND WAITS)
ENDPROG
```

A program is constructed from blocks. A block is one single line in a text editor, e.g., everything you type on a line until your press ***Enter***. Since the size of the screen is limited on the HP-100 itself, the program box supports wrapping, and each block must be terminated with the End of Block character, ***;*** (semicolon). Now, take a detailed look at each block of the program listed above.

BLOCK 1: `PROG1 (A SAMPLE PROGRAM)`

A program will not run if the first block is not properly constructed. The first identifier, **PROG1,** is the name of the program. A valid name must begin with a letter (A-Z), and only contain letters, numbers, or

the underscore character. Sample valid program names are:

- PROG1
- HP_100Bend
- P1_18_2012

It is recommended that the name of the program matches the filename used to store the program to the hard disk or a USB key. Any new program created on the press follows this formatting suggestion.

The next portion surrounded by parenthesis is a special comment, reserved for the description of the program. It will appear in the description text box on the ***Program Operations*** tab. From then on, the construction of the program is up to the operator.

Block 2 is an empty block so it is skipped. Empty blocks are useful to make the program easier to read, and are completely optional between operation blocks.
Continue looking at the next block.

BLOCK 3: `(THIS IS A COMMENT)`

Any text that is surrounded by parenthesis , **(** and **)**, is a comment and completely ignored by the controller.

BLOCK 5&6:
```
RC=-0.050 (SET A RAM CORRECTION HINT)
RC=AUTO (LET THE RAM AUTO CORRECT FROM NOW ON)
```

Comments can take up the entire line, or be embedded in a line as demonstrated here. The ***RC*** command sets the **r**am **c**orrection value for machines without a proportional valve. Just as described earlier, the recommended way of using ram correction is to supply a hint, and then allow the ram to automatically correct itself, done so by specifying the ***AUTO*** value.

BLOCK 8: `RT=5.0 RRM=4.8 (EXTEND THE RAM AND RETRACT)`

This is the first block in which ram movement takes place. The ***RT*** command specifies a **r**am extension **t**arget. The ***RRM*** command specifies a **r**am **r**eturn point that is **m**odal. This tells the controller to imply an ***RR*** command on any line that contains an ***RT*** command. The purpose of this command is to reduce the amount of typing necessary if the ram should always return to the same clearance point. ***RRM*** remains in effect unless it is overridden in a block by the ***RR*** command, or it is canceled by assigning the ***OFF*** value.

BLOCK 10: `TEXT="DO SOMETHING IMPORTANT"`

Block 10 prints a user message to the textbox located in the bottom of the screen when the press is running a program. ***TEXT*** is one of the "string" based commands, which means any assignment is required to surrounded by quotations ***", ".***

BLOCK 12: `RT=5.1 (EXTEND AND RETRACT THE RAM TO THE RRM)`

Block 12 extends the ram to the target point, AND retracts it to the previously specified ***RRM*** point.

BLOCK 14: `RT=5.125 RR=4.5 REPEAT=5(OVERRIDES RRM)`

Block 14 extends the ram to the target point, but overrides the ***RRM*** point and returns to the location specified by the **r**am **r**eturn command, ***RR***. An ***RR*** command always has priority when retracting the ram. Also introduced on this block is the ***REPEAT*** command, which simply repeats the current block the specified number of times.

BLOCK 16: `RT=5.050 (EXTEND AND RETRACT THE RAM TO THE RRM)`

Block 16 extends the ram to the target point and retracts again to the ***RRM*** specified previously. This is included to demonstrate that the ***RRM*** is still valid after being overridden in the block before.

BLOCK 18-21:
```
RRM=OFF (TURN OFF THE RRM POINT)

RT=5.25 (EXTENDS THE RAM BUT DOES NOT  RETRACT)
RR=1.0 (RETRACTS THE RAM)
```

Block 18 turns off the modal ram return point. Now, the press requires an explicit ***RR*** command to retract the ram, as displayed in blocks 20 and 21. The ***RT*** command in block 20 will stop when the ram has finished extending and wait, while the ***RR*** command in block 21 will retract the ram and wait.

BLOCK 24: `ENDPROG`

Block 24 rewinds the program through the ***ENDPROG*** command. It then causes a reset, and the controller is ready to run the program again after the operator presses the ***Start*** button. If this command is omitted, then the program will automatically rewind to the beginning and start again without the need for the operator to press the ***Start*** button.

Summary of Automatic Programming Commands

RT
Sets a ram target point.

Example:
```
RT=5.25
```

DWELL
Pauses the ram at the extension point for the specified number of milliseconds before retracting.

Example:
```
RT=5.25 DWELL=500 (PAUSE FOR 1/2 SECOND AT RT POINT)
```

RR
Sets a ram return point. It will override any active ***RRM*** command.

Example:
```
RR=1.0
```

RRM
Sets a modal ram return point. After this command, any line that contains an ***RT*** command will automatically return to the specified ***RRM*** point, unless it is overridden by an ***RR*** command.

Example:
```
RRM=4.0
```

REPEAT
Causes the block to be repeated specified number of times. Incrementing the step counter each time.

Example:
```
RT=5.125 RR=4.5 REPEAT=5
```

TEXT
Prints a user specified message to the automatic program message box. The value must be surrounded by **"**, **"** since it is a string command.

Example:
```
TEXT="DO SOMETHING IMPORTANT"
```

SOUND
Plays a user specified ****.wav*** file. ***SOUND*** is a string command which means the value must be surrounded by ***", ".*** The file must be in the same directory as the program, or in one of the search directories. It is always valid to place a sound file in the ***/Jd2User/CommonSounds*** folder found on the

press hard disk. The other search path is a folder one directory above the program *.txt file, also named ***CommonSounds***. To clarify the following file structure shows valid sound file locations.

On the press

- *Jd2User*
  - *CommonSounds*
    - *MYSOUND1.WAV*
  - *PROG1*
    - *PROG1.TXT*
    - *MYSOUND2.WAV*
  - *PROG2*
    - *PROG2.TXT*

This is the recommended directory structure for programs stored on the press. It serves to organize multiple sound files that can be attached to a program much cleaner than dumping all files into a single directory. In this example, ***PROG1*** can play both ***MYSOUND2.WAV***, and ***MYSOUND1.WAV***, while ***PROG2*** can only play ***MYSOUND1.WAV***. If you create a program through the ***New*** button on the ***Program Operations Tab***, it will generate the program nested in a folder as described above. The proper file structure for a USB key is described below.

On a USB Key

- *USBRootFolder*
  - *CommonSounds*
    - *MYSOUND1.WAV*
  - *PROG1*
    - *PROG1.TXT*
    - *MYSOUND2.WAV*
  - *PROG2*
    - *PROG2.TXT*

This is the recommended directory structure for programs stored on a USB key. It serves to mimic the structure found on the press hard drive. ***USBRootFolder*** can be any directory, but it is recommended to put all programs and a ***CommonSounds*** folder inside of it as described. In this example, ***PROG1*** can play both ***MYSOUND2.WAV***, and ***MYSOUND1.WAV***, while ***PROG2*** can only play ***MYSOUND1.WAV***, just as in the ***On the Press*** example.

Example:

```
SOUND="MYSONG.WAV" (PLAYS A FILE NAMED MYSONG.WAV)
```

ENDPROG

Rewinds the program and forces a reset.

(...)

Anything surrounded by parenthesis is a user comment. Ignored by the controller.

Example:

```
(THIS IS A COMMENT)
```