Divelbiss Bear Bones

Training Tutorial

This tutorial seminar will not replace all the information in our programming manual (ICM-UM-05). This training tutorial will introduce you to the programmer (ICM-PR-05) so that you can take full advantage of the manual.

We do provide customer assistance. Divelbiss Corporation does not have complete data concerning the use or application of the products described herein. Divelbiss Corporation cannot assume any responsibility for the application, product or patent infringements which may result from such assistance.

Divelbiss Corporation
9778 Mount Gilead Road
Fredericktown, Ohio 43019
Telephone: 800/245-**BEAR**(2327)
Local: (740) 694-9015
Fax: (740) 694-9035
Website: www.divelbiss.com
E-Mail: divelbiss@divelbiss.com
10/27/98

REV. A

WARNING

The Bear Bones PLC, as with other solid state devices, must not be used alone in applications which would be hazardous to personnel in the event of failure of the device. Precautions must be taken by the user to provide mechanical and/or electrical safeguards external to this device. This device is NOT approved for domestic or human medical use.

Table of Contents

I. RELAY LOGIC VERSUS LADDER LOGIC

A comparison of a hard-wired relay logic system and a system consisting of any of the Bear Bones PLCs will show you the similarities which make the Bear Bones programming language easy to apply. Figure 1.3 shows a hard-wired relay control logic system. For purposes of comparison, it is divided into sections, consisting of:

Input Devices, including devices operated manually (push buttons) and devices operated automatically (limit switches) by the machine or process being controlled.

Relay Control Logic, consisting of relays interconnected to energize or de-energize output devices in response to the status of the input devices, and in accordance with the logic designed into the control circuit.

Output Devices, consisting of motor starters, solenoids, etc. which control the machine or process.

In place of hard-wired relay circuitry, the Bear Bones controller contains programmed instructions which use **relay-type symbology**. Thus, with the Bear Bones controller, you are still employing familiar relay logic methods-but in a new way, which provides a lot more flexibility.

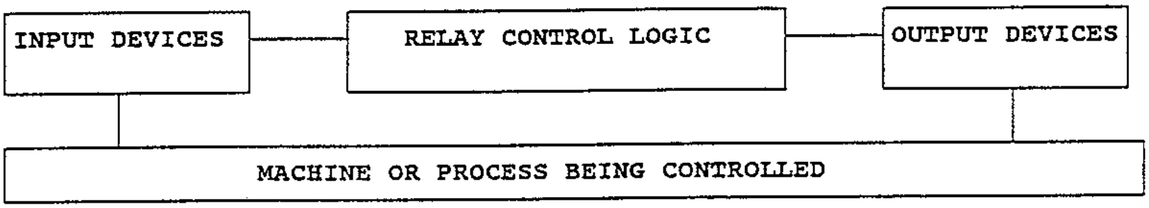
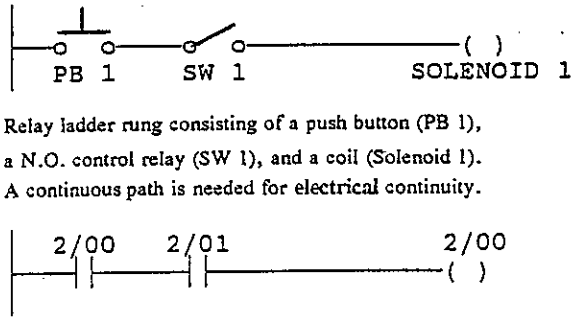


Figure 1.3

The simplicity of the ladder diagram programming format is illustrated in Figure 1.4, which shows a relay ladder rung as used in relay control systems and a similar ladder rung as used in programming the Bear Bones PLCs.



Relay ladder rung consisting of a push button (PB 1), a N.O. control relay (SW 1), and a coil (Solenoid 1). A continuous path is needed for electrical continuity.

Figure 1.4

Similar PC ladder rung. Instruction address numbers 2/00, 2/01, and 2/00 are identified with PB 1, CR 1, and CR 2 respectively. A continuous path is needed for logical continuity.

BEAR BONES ADDRESSING STRUCTURE

Binary Addressing Structure

The basic structure of any digital circuit utilizes the binary number system. Any digital output can have one of two possible binary digits: "0" and "1" - more commonly called a "BIT". Each bit location has a corresponding decimal value assigned to it. These binary bit locations are labeled on the expander boards with their decimal equivalent directly above each location. Each of the three (3) address selector pins are assigned a Binary Coded Decimal(BCD) value of 4, 2, or 1. Converting binary numbers to their decimal equivalent is quite simple. Please follow the example below.

Example: The binary number 0 1 1 would represent a decimal value of 3.

Look at the right half of the table and locate the row with the bits 0 1 1. These binary bits represent a decimal value of 4, 2, and 1 respectively. The number 0 1 1 has a "1" in positions 1 and 2, so the conversion is made by adding these values 1 + 2 = 3.

Decimal Value = Pg #		4	2	1	
0		0	0	0	
1	Do Not Use	0	0	1	Reserved for CPU's
2		0	1	0	
3		0	1	1	
4		1	0	0	
5		1	0	1	
6		1	1	0	
7		1	1	1	

Table 1.1

Our paging systems works similar to this pattern. Each Bit location on an expander board will have a jumper that needs to be removed or placed to access the correct page wanted. **To address an I/O expander board, place a jumper on the corresponding bit where a "0" is located in the table.** See page 5 (Address Selector Locations) for proper address selector identification for your particular expander board.

Example: Address an expander board to Page 5. Using Table 1.1, we can see that page 5 is represented in binary as 1 0 1. In this case, you would need to place a jumper in the 2s position(middle position) to represent the "0" in the binary number.

Since each page is made up of 16 inputs and 16 outputs, each expander board takes up half of a page (8 inputs and 8 outputs) except for the ICM-HDIO-01P,02P,06P,07P which take a whole page. This means that we have an upper and a lower part of the page. You will notice another jumper located on the expander board signified by the letters "U" and "L". "U" denotes the Upper part of the page and "L" denotes the Lower part of the page. Consequently, two (2) expander boards can be used on one(1) page. One on the upper half and one on the lower half. The addresses are entered as **PAGE #/INPUT# or OUTPUT#**. The lower half of the page has I/O addresses corresponding to X/00 through X/07. The upper half of the page has I/O X/08 through X/15 (X signifying the page numbers).

Example: To address an expander board at Page 4 LOWER, place a jumper in the 2s and 1s position and a jumper connecting the middle pin to the "L" pin on the Upper/Lower selector.

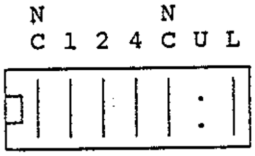
The addresses available to use on this expander board are as follows:

INPUTS	4/00 through 4/07
OUTPUTS	4/00 through 4/07

ADDRESS SELECTOR LOCATIONS

Bear Bones Expanders (Includes Baby Bear) PIC or ICM-IO-XX

The standard Bear and Baby Bear Bones expanders (Non-High Density) have a 14-pin female address selector socket located on the expander board. The 14-pin male plug contains gold colored jumpers used to select the correct address. The expander boards are shipped addressed at Page 0 LOWER. There are also extra jumpers that are included. These extra jumpers are labeled as NC or No Connection.



```
NC = No Connection (Extra Pin)
 1 = Bit 1
 2 = Bit 2
 4 = Bit 4
NC = No Connection (Extra Pin)
 U = Upper I/O  8 thru 15
 L = Lower I/O  0 thru 7
```

HIGH DENSITY EXPANDERS ICM-HDIO-XX

The standard High Density expanders have 4 blue colored jumpers that are removable located on the expansion board. The expander boards are shipped addressed at Page 0 LOWER.

```
                  ADDRESS
                  SELECTOR
                    4 2 1
    L .             . . .
      .             . . .
    U .
```

CPU ADDRESSING

Each CPU has its own I/O associated with it (Except for High Density CPU's). Following is a list of part numbers for our PLC's that accommodate their own I/O:

ICM-BB-12 thru BB-110 (Includes Bear and
PIC-BB-14 thru BB-22 Baby Bear Bones)

BEAR BONES PLC

The standard Bear Bones PLC's have a total of 8 inputs and 8 outputs associated with the main board available with different types of inputs and output voltages. The page addresses for these eight (8) I/O are pre-assigned to PAGE 1 UPPER (1/08 through 1/15). This address cannot be changed. PAGE 1 LOWER is reserved for the CPU's use and the Cub Expanders and Special Function Card. The following table lists the I/O map for the Bear Bones PLC's.

BABY BEAR BONES PLC

The standard Baby Bear Bones PLC's have a total of 5 DC inputs and 5 relay outputs associated with the main board. The page addresses for these five (5) I/O are pre-assigned to PAGE 1 LOWER (1/03 through 1/07). This address cannot be changed.

HIGH DENSITY BEAR BONES PLC

The standard High Density Bear Bones PLC's have NO external I/O associated with the main board. All I/O is accessed by the separate expander boards. All standard Bear Bones Expanders are compatible with the High Density PLC.

BEAR/BABY/HD BEAR BONES I/O MAP

INPUTS	OUTPUTS
Page 1	
00 - DO NOT USE	00 - DO NOT USE
01 - DO NOT USE	01 - DO NOT USE
02 - WATCHDOG led (10 Hz Time Base)	02 - DO NOT USE
03 -	03 -
04 - (Inputs and Outputs 03 through 07 can be used	04 -
05 - with the Baby Bear, Cub Expander, TTL/CMOS,	05 -
06 - Expanders, and Special Function Cards)	06 -
07 -	07 -
08 - (Inputs and Outputs 8 through 15 are the inputs	08 -
09 - and outputs available on the Bear Bones CPU board)	09 -
10 -	10 -
11 -	11 -
12 -	12 -
13 -	13 -
14 -	14 -
15 -	15 -
Page 2	
00 -	00 -
01 - (Inputs and outputs 00 through 07 are used	01 -
02 - by the first expander)	02 -
03 -	03 -
04 -	04 -
05 -	05 -
06 -	06 -
07 -	07 -
08 - (Inputs and outputs 08 through 15 are used	08 -
09 - by the second expander)	09 -
10 -	10 -
11 -	11 -
12 -	12 -
13 -	13 -
14 -	14 -
15 -	15 -

Table 1.2

Ladder Logic Symbols

Symbol -] [-
Name: Normally open contact

The normally open contact is programmed when the presence of the referenced signal is needed to turn an output ON. When evaluated, the referenced address is examined for an ON, or TRUE condition. The referenced address may represent the status of an external input or internal output. If, when examined, the referenced address is ON, then the normally open contact will close and allow logic continuity (power flow). If it is OFF, or FALSE, then the normally open contact will assume its normal programmed state (open), thus breaking logic continuity (power flow).

Symbol -]/[-
Name: Normally closed contact

The normally closed contact is programmed when the absence of the referenced signal is needed to turn an output ON. When evaluated, the referenced address is examined for an OFF or FALSE, condition. The referenced address may represent the status of an external input or an internal output. If, when examined, the referenced address is OFF, then the normally closed contact will remain closed allowing logic continuity (power flow). If the referenced address is ON, or TRUE, then the normally closed contact will open and break logic continuity (no power flow).

Symbol -( )-
Name: Energize coil

The energize coil instruction is programmed to control either an output connected to the controller or an internal (control relay) output. If any rung path has logic continuity, the referenced output is turned ON. The output is turned OFF if logic continuity is lost. When the output is ON, a normally open contact of the same address will close, and a normally closed contact will open. If the output goes OFF, any normally open contact will then open, and normally closed contacts will close.

Symbol -(/)-
Name: De-energize coil

The de-energize coil instruction is programmed to control either an internal output or an output device connected to the controller. If no rung path has logic continuity, the referenced output is turned ON. The output is turned OFF if logic continuity is achieved. When the output is ON, normally open contacts of the same address will be closed, and normally closed contacts will be open. If the output goes OFF, normally open contacts will open, and normally closed contacts will close.

II. GETTING STARTED

The first thing you must do is furnish power to the PR-05. Please refer to Figure 1.4. There are three receptacles on the top of the PR-05. The power supply connects to the center connector. Look at the power supply (ICM-PS-15). There is a standard three prong grounded plug for insertion into a standard three hole wall socket. The other end of the power supply cable will fit into the power port if you insure that the arrow on the metal plug points up.

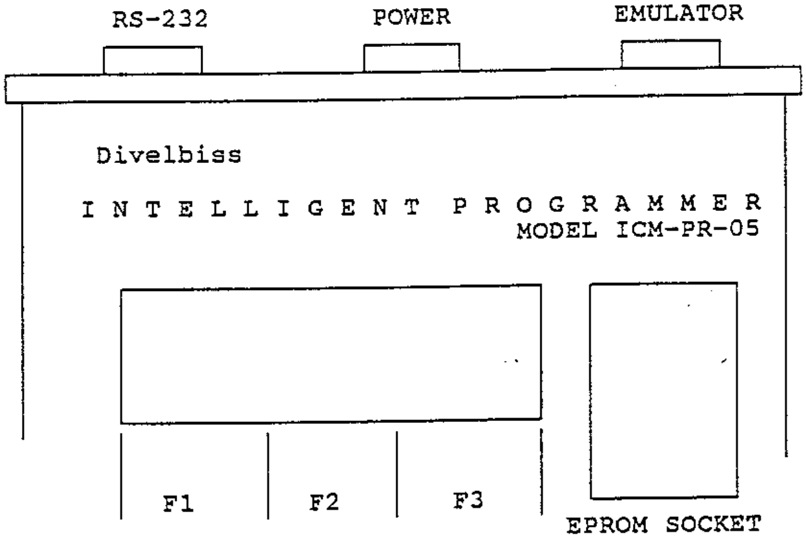


Figure 1.4

Plug the power supply into the PR-05. Now plug the power supply into any 120V ac outlet. The display will read:

PR-05 VER 1.82
Programmer Ready

Press <ENTER>

This prompt indicates that the programmer is ready.

Note: The message on your PR-05 may very depending on the version of software you are using. The message shown here reflects the most current version available.

Press the **ENTER** key. After pressing the ENTER key, the main menu will appear on the screen. You will notice that the function keys (F1, F2, F3) correspond to the appropriate menu title. The **ADV** key will scroll the menu through different options that are available under the Main Menu screen. Pressing the **ADV** key enough times will eventually return you to Prog Print Run. At any time, you can press the **Menu Return** key to get back to the Main Menu.

III. RUNG FORMAT

The Bear Bones ladder diagram consist of rungs. Rungs are referred to the basics of ladder logic. Each rung has a format of an 8x8 element matrix. This means that in each rung, there are 8 horizontal lines of logic that can be programmed with 8 elements in each line. The first line of the rung is labelled as ".0" the second labelled as ".1" sequencing in this format up to ".7". This gives us a total of 8 horizontal lines in each rung.

Each line of the rung can contain up to 8 elements. An element being a normally open or normally closed contact, control relays, etc. Figure XXX shows how each rung is divided into 8 lines of logic with elements in each line.

```
                          RUNG 001

.0     ---[ ]---[/]---[ ]---[/]---[/]---[ ]---[ ]---(   )

.1     ---[/]---[ ]---[/]---[ ]---[ ]---------------(   )

.2     ---[ ]-+-[/]---[ ]---[/]---[ ]---------------(   )
              |
.3     ---[/]-+

.4     "

.5     "

.6     "

.7     "
```

IV. PROGRAMMING

The PR-05 is used to program the Bear Bones and Baby Bear in ladder diagram notation. The PR-05 software guides the user through the ladder by always moving to the next location to be programmed - left to right, top to bottom. The cursor can be positioned anywhere within the ladder using the arrow keys: **→ (right), ← (left), ↑ (up), ↓ (down).**

These programs will help you get used to some of the keyboard elements. We want you to understand how to draw your ladder diagram. Then we will cover the more powerful instructions.

At the Main Menu - Prog Print Run, press the **F1** key under **Prog.**

If there is a program stored in the memory, the PR-05 will ask if you want the old programmed cleared or to start a new one. Press F3 for new. (note: If "Clear Program?" appears on the display, press yes) you are now ready to enter your first program.

1. Simple Rung

This program will consist of one input and one output.

Press -] [- (Normally Open Contact)
The PR-05 is now requesting an INPUT address.
Press **2,0,0, Enter**. The cursor has moved to the right of the contact. This means that this element is programmed. A normally open input contact with an address of 2/00.

Display

```
  2/00
-] [-▒
```

Next press -( )- (Output Coil)
The PR-05 will place the coil on the right side of the screen and draw the connecting ladder for you.
The PR-05 is now requesting an OUTPUT address.
Press **2,0,0, Enter**. The cursor is now under the normally open contact.

```
EXAMPLE:      2/00                       2/00
              -] [-------------------( )--

                ▒
```

The PR-05 will allow you to continue to program on rung one, or complete rung one by pressing **ENTER**. Remember that this rung can be changed later.

This rung would be read "When input 2/00 is true, output 2/00 is true."

Use the cursor control keys (F1, F2, F3, and ADV) to place the cursor over the input contact element. Notice the identification information at the bottom of the screen. Do the same for the coil.

A common error is to forget to press ENTER after typing the address only to find that the PR-05 "beeps" and will not allow further programming. Just press ENTER to cure.

Press **ENTER** to go to Rung 2.

Press **Menu Return**. Note: If you do not press enter to go to Rung 2, you will not be able to leave the edit mode. If you press Menu Return before pressing ENTER, the PR-05 will give you an "Abandon Changes?" "YES" or "NO" message. If you type yes, the PR-05 will return you to the main menu without storing your program. If you type no, the PR-05 will re-enter the edit mode to allow you to press the ENTER key to store your ladder program. Pressing the ENTER key will allow the PR-05 to store the ladder logic that you have just programmed. In doing this, the PR-05 will eliminate the extra programming lines in the rung that you have not used. This saves on memory space where the program is stored. Then press Menu Return.

At the main menu...

```
MAIN MENU

Prog  Print  Run
```

Press **Run** (F3) the PR-05 will briefly display:

For the ICM-IF-BB:

```
 Downloading
Program to PLC
   BB-XX
```

For the PIC-EM-02:

```
 Downloading
Program to PLC
  rHD-16K
```

Your actual display will depend on the version of software and the PLC type you are using. For software versions 1.69 and lower only the ICM-IF-BB display is possible. If you are using version 1.70 or higher you could get either display depending on the PLC type you are using.

Note that the XX represents the type of Bear Bones you are programming. Refer to the data sheet furnished with your controller to make this distinction. This configuration must be correct in order for the program to run correctly. Also, see the ICM-UM-05 programming manual.

After downloading is complete, (the amount of time will very depending on the length of your program), the PR-05 will display:

```
      Rung 001

-] [----------( )-
RESET  RUN   SCAN
```

At this point you have three options. If you choose RESET, the PR-05 will reset the outputs. If you choose RUN, the PR-05 will run the program on a continuous scan. If you choose SCAN, the PR-05 will scan the program once. Press **Run** (F3). The Watchdog Led on the Bear Bones board should be flashing at a steady rate. If you turn on input 2/00, output 2/00 should activate. If you look at the display on the PR-05 you will notice that the elements in your rung are flashing, this is telling you that there is power flow in these elements. If you turn off input 2/00, notice that the elements stop flashing. As you can see, this could be a very helpful tool in troubleshooting your program.

2. Series Rung (CLEAR OLD PROGRAM) (AND)

This program will consist of 2 inputs "anded" together to activate one output.

Enter the first contact by pressing **-] [-,2,0,0, ENTER.** The cursor is to the right of the contact.

Press -] [-, 2,0,1, ENTER. Now the cursor is to the right of the second contact.

Press -( )-,2,0,0, ENTER.

```
                          Rung 001

        EXAMPLE:       2/00 2/01                2/00
                      -] [---] [----------------( )-
```

The cursor is now under the first contact element. This line of logic can be read "If input 2/00 AND input 2/01 are true, then output 2/00 is energized".

Press ENTER to go to rung 2.

You can now run this program.

Inserting Series Contacts

For editing purposes, insert a normally open contact between input 2/00 and 2/01. The PR-05 will insert a space left of the position of the cursor.
Position the cursor on contact 2/01.
Press the **Function** key.
Press **F3 (Insert)**.
Press **F1** (Series). This will open a space directly to the left of the cursor.
Press -] [-, 2,0,2.

This line of logic can be read as "If input 2/00 AND input 2/02 AND input 2/01 are true, then activate output 2/00.

```
 2/00   2/02   2/01           2/00
-] [---] [---] [----------( )-
```

Press **ENTER** to go to Rung 2.
You can now RUN this program.

3. Parallel Rung (CLEAR OLD PROGRAM)   (OR)

This rung will consist of two inputs "ored" together with one output.

Press -] [-,2,0,0, ENTER. Now press the " ┬ " (branch down key). The cursor is now to the right of the branch down element.

Enter the output by pressing -( )-,2,0,0, ENTER. The cursor is now under the -] [- 2/00.

Now enter -] [-,2,0,1, ENTER. Press the " ┘ " (close branch key). This rung can be read "If input 2/00 OR input 2/01 is true, then output 2/00 is energized."

```
                              Rung 001
        EXAMPLE:           2/00                     2/00
                          -] [-+------------------( )-
                               |
                          2/01 |
                          -] [-+
```

Press **ENTER** to go to rung 2.
You can now run this program.

Inserting Parallel Contacts

Position the cursor directly over input contact 2/01.
Press **Function, F3 (Insert), F2 (Parallel)**. This will open a space directly above the original position of the cursor allowing us to insert a contact above input 2/01.
Press **-] [-,2,0,2, ENTER**
Press ⊣ . This will complete the branch up.

```
            EXAMPLE       2/00                     2/00
                          -] [-+------------------( )
                          2/02 |
                          -] [-+
                          2/01 |
                          -] [-+
```

This rung can be read "If input 2/00 OR input 2/02 OR input 2/01 are true, output 2/00 will energize.

4. Control Relays (CR's) (CLEAR OLD PROGRAM)

Control Relays are used to perform logic functions. They can also be used as an extension of an input. Control relays are similar to real world outputs in that once turned on, they remain on until turned off. Since a CR in the ICM product line is just a bit or a single memory location, the contact and the coil are always in the same state. That is, if the coil of CR-001 is true, the contact of CR-001 is also true.

Press **-] [-,2,0,0, ENTER** to select an input.
Press **-( )-, CR,1, ENTER** to name the control relay that is to be controlled by input 2/00. Contacts either normally open or normally closed can be used anywhere in the program and as often as you require from this CR.
Press **-] [-, CR,1, ENTER** to select contacts from CR-001 coil.
Press**-( )-, 2,0,0, ENTER** to select a real world output.

```
                                Rung 001

        EXAMPLE:             2/00                   CR-001
                             -] [------------------( )-

                             CR-001                  2/00
                             -] [------------------( )-
```

When input 2/00 is closed, CR-001 coil will energize, this will energize input CR-001 and output 2/00 will energize. When input 2/00 is opened, CR-001 is de-energized which de-energizes output 2/00. Remember, this is a PLC and it is scanning through its program. We will look a little deeper into this program to see what actually happens.

Assume input 2/00 is on, and output 2/00 is on. When we turn off input 2/00, the output seems to go off at once. Due to scan time, there is a measurable amount of time between events.

NOTE: As a fast rule of thumb you can figure that your total scan time for a program will be 5 milliseconds per 1000 bytes.

Press **ENTER** to go to Rung 2.

You can now run this program.

V. SPECIAL FEATURES

1. FIND

The simple insert adds a contact to a previously completed line of lógic ahead of the output instruction. At this time we will illustrate the search feature of the PR-05 as well.

Press the **FUNCT** key, the PR-05 displays:

```
Prog Functions

Fun  Instr  Insert
```

Press **ADV** to display:

```
Prog Functions

Del  Find   Next
```

Now press **FIND** to display:

```
Find Options

In  Out  Cr
```

Now press **ADV** to display:

Find Options

Rung Fun Instr

Now press **Rung, 1, ENTER.** The cursor should now be on Rung 1. You can start your program at this point. This may seem pointless in this example, however, you can see how you might search for something in your program without looking through your the entire logic. Another option is the NEXT choice on the menu. NEXT allows you to search for the same element in the program and see if you have used it more than once. This can be a very helpful tool in troubleshooting your program.

2. MEMory

While programming with the PR-05 it is possible, at any time, to determine how much memory is being consumed by your program. In doing this, the PR-05 will also tell you how many bytes are being used and how many bytes are free. The PR-05 will also display the minimum size EPROM needed to store the program.

While in the programming mode:

Press **FUNCT, ADV, ADV, MEM or F3.** This will display the memory information.

EXAMPLE - Seal Circuit

```
2/00      2/01                   CR-001
-]/[---+---] [---+----+---------( )
       |         |    |
       | CR-001  |    |          2/00
       +---] [---+    +---------( )
```

Upon trying to store this program, a "Contact Missing" error will occur. The PR-05 will position the cursor at the point where the problem occurred which is at the third branch down connection. With this example, the PR-05 will not recognize this as a valid entry. The following program is the corrected solution to the error.

```
2/00      2/01                   CR-001
-]/[---+---] [---+--------------( )
       |         |
       | CR-001  |               2/00
       +---] [---+--------------( )
```

The above example shows an application how Control Relays can be used. This is a simple "Seal" or "Motor Starter" circuit. If input 2/01 is true, this energizes CR-001 and output 2/00. Since CR-001 is in parallel with input 2/01, input 2/01 can drop out and the output remains energized. To reset the circuit, energized input 2/00 to de-energize CR-001 which de-energizes output 2/00.

VI. PROGRAMING FUNCTIONS

The following are examples of functions (counters, timers, drums, etc.). We will demonstrate how to enter each function. All input logic will be kept simple to concentrate on programming the functions.

There are 32 programming functions available for use as timers, counters, etc. in the Bear Bones line of controllers. The Baby Bear gives you the option for 10 or 32 programming functions. The Baby Bear is preset for 10 functions. If you decide you need more than 10 functions, a jumper on board will need to changed which allows for 32 functions; however, **if you configure the Baby Bear for 32 functions, Output 1/03 will be sacrificed.**

There are a total of 491 control relays available to the programmer. There are 107 free control relays that are available (CR-001 to CR-107). Each function programmed will use 12 internal control relays starting at CR-108. The first function programmed will use a group of 12 control relays starting at CR-108 to CR-119. The second function programmed will use the next avaiable group of 12 control relays starting at CR-120 to CR-131 and so on.

The PR-05 will keep track of the amount of functions programmed and default to the next available group of control relays.

Note: A Drum function will automatically use two (2) groups of 12 control relays for its memory.

1. TDPU - Time Delay on Pick Up (CLEAR OLD PROGRAM)

Press -] **[-,2,0,0, ENTER**. Input 2/00 has been selected as our input address.

Press the **FUNCTION** key to display the function sub menu:

```
Prog Functions

FUN  INSTR  INSERT
```

Now press **FUN** to display the special functions sub menu:

```
Spec Functions

TDPU  TDDO  RET
```

Since we are programming a TDPU timer, press **F1 or TDPU**. The PR-05 defaults to the lowest number for the TDPU that is not programmed. In this case it is number 1, so press **ENTER**. The PR-05 suggests function group 1 for you, so again press **ENTER**. You may assign function group numbers in the range of 1 to 32, but usually it is best to just let the PR-05 choose the lowest number available.

The PR-05 is asking you to select a time base. There are four available, you must press ADV to display the fourth. In this example we will select **0.1** seconds time base.

The PR-05 is now asking you to select a setpoint type, press **F1 or INTERN**al. This means that you will enter a setpoint from the keyboard and it will be fixed in software. For an internal setpoint, you may select any number from 1 to 255. For this example press **5,0,ENTER**. You have chosen to time to 50 "ticks" each 0.1 seconds long, giving you a setpoint of 5.0 seconds with a resolution of 100 mSec.

You may now use contacts from TDPU-1. Either normally open or normally closed. Press -] [-, **FUNCT, TDPU**. The PR-05 will default to the next available TDPU, which is 2; however, we are still working with TDPU-1.

Press **1, ENTER**
Press -( )-,**3,0,0, ENTER**. Output 3/00 has been assigned as the output to use.

```
EXAMPLE:                     Rung 001
          2/00                                  +--------+
          -] [-----------------------------------| Reset  |
                                                | TDPU 1 |
                                                |  5.0s  |
                                                +--------+

           TDPU01                                  3/00
          -] [---------------------------------------( )-
```

Press **ENTER** to go to Rung 2.

You may now **RUN** this program.

When input 2/00 is true, TDPU01 starts timing. At the end of the time period (5.0 seconds), the contact from TDPU01 closes and output 3/00 energizes. The timer can be reset by making input 2/00 false.

Place the cursor on each of the coils of this timer and note the identification. When the system is running, you can see the timer count up and reset. Once programmed, contacts from this timer, like all functions, may be used wherever required in the program, as often as you need it.

2. TDDO - Time Delay on Drop Out (CLEAR OLD PROGRAM)

Press **-] [-,2,0,0, ENTER**. Input 2/00 has been selected as our input address.

Press **FUNCT, FUN, TDDO**. Note the default is #1 since this is the first TDDO in this program. Press **ENTER**. The PR-05 selects function group 1 for you, press **ENTER**.

Again, select a 0.1 second time base to keep the run time down to a reasonable amount of time.

The PR-05 wants you to select where it will read the setpoint from.
Press **INTER**Nal. This means that the setpoint will be preset internally.
Press **5,0, ENTER**. This will preset the time to 5.0s.

You may now use contacts from TDDO-1, either normally open or normally closed.

Press **-] [-,FUNCT, TDDO, 1, ENTER.**
Press **-( )-, 3,0,0, ENTER.**

```
EXAMPLE:               Rung 001
         2/00                        +--------+
         -] [------------------------| Reset  |
                                     | TDDO 1 |
                                     |  5.0s  |
                                     +--------+
          TDDO 1                       3/00
         -] [----------------------------( )-
```

When input 2/00 is closed output 3/00 is energized. When input 2/00 is opened (Drops Out), the timing sequence starts, output 3/00 remains on. At the end of the preset value, output 3/00 turns off.

Press **ENTER** to go to Rung 2.

You may now run this program.
Put the cursor on the second coil of this timer to watch the timing sequence.

3. RET - Retentive Timer (CLEAR OLD PROGRAM)

Press **-] [-,2,0,0, ENTER** to select an input.

Press **FUNCT, FUN, RET**. The default is 1, press **ENTER**. The function group is 1, press **ENTER**. Select the **0.1** second time base.

Select the **INTER**Nal setpoint and use **65** as your number, press **ENTER**. Your setpoint is now 6.5 seconds.

Note that the cursor is opposite the second coil of the timer.
Press -] [-,2,0,1, ENTER. Now press the "-" (dash) key and hold it down until the "beep" is heard and the setpoint is displayed.

Press --> to position the cursor below the input 2/01.
Press -] [-,FUNCT, RET, 1, ENTER.
Press -( )-,3,0,0, ENTER.

When input 2/00 is closed, the retentive timer is enabled. Close input 2/01 to run the timer. If input 2/01 is opened, the timer will stop but not reset. Re-close input 2/01 and the timer will start to time again where it had stopped. When the setpoint is reached, output 3/00 will energize.

```
EXAMPLE:                    Rung 001
        2/00                                     +--------+
        -] [-------------------------| Reset  |
                                                  | RET 1  |
        2/01                                     | 6.5s   |
        -] [-------------------------|        |
                                                 +--------+

        RET 1                                    3/00
        -] [----------------------------( )-
```

4. PGEN - Pulse Generator (CLEAR OLD PROGRAM)

Press -] [-,2,0,0, Enter to select an input.

Press FUNCT, FUN, ADV, PGEN, ENTER, ENTER. Now press ADV and select the .1 sec time base. Press INTERNal and use 20 as your setpoint number. Press ENTER.

Press -] [-, FUNCT, PGEN, 1, ENTER.
Press -( )-, 2,0,0 ENTER.

When input 2/00 is closed, the pulse generator will begin timing. At the end of the time period (2 sec.) output 2/00 will turn on for one scan. The pulse generator will reset and start timing again and repeat the same operation.

```
EXAMPLE:                    Rung 001
        2/00                                     +--------+
        -] [-------------------------| Reset  |
                                                  | PGEN 1|
                                                  |  2.0s  |
                                                 +--------+

        PGEN 1                                   2/00
        -] [----------------------------( )-
```

Contacts from this pulse generator can be used to drive anything which may require a constant time base such as a drum sequencer, counter, etc.

These numbers indicate which element of the matrix the cursor is on. The step position is first and is followed by the channel number.

Press the **1** key. Note that the cursor has moved to channel two. Also note that the position indicator displays 1 x 2. This indicates Step 1 Channel 2. Now press **0**. When you run this drum, Channel 2 will be off in Step 1. Press **0**. You are in Step 1 Channel 3. Channel 3 will be off in Step 2. Continue programming until the following drum matrix is programmed as in Figure 3.

Any channel can be enabled (programmed with a 1) in any step. All channels can be enabled in any or all steps.

```
EXAMPLE:                            CHANNEL
                                     12345
                          S|   1.  10000
                          T|   2.  01000
                          E|   3.  00100
                          P|   4.  00010
                          S|   5.  00001
```

Figure 3

Press **ENTER.**

Press **-] [-,2,0,1, ENTER.** Press and hold the "-" key until a "beep" is heard and the step 5 x 5 is displayed.

You may now program contacts of the drum in Rung 1 or Rung 2 either normally open or normally closed. Each channel may be programmed anywhere in the program and as often as needed. Each contact is identified by its drum number and channel number.

```
EXAMPLE:                       Rung 001
         2/00                                    +--------+
         -] [-------------------------------|Reset   |
                                                 | DR 1   |
         2/01                                    | 5X5    |
         -] [-------------------------------| STEP   |
                                                 +--------+
```

When input 2/00 is closed, the drum will be enabled. The drum will be in Step 1 until input 2/01 is closed advancing the drum to Step 2.

CAUTION! It is easy to forget that the drum's home position is Step 1. This is the position where the drum returns when it is reset. Use care in programming any channel to be ON in Step 1.

Press **ENTER** to go to Rung 2.

5. CNTR - Counter (CLEAR OLD PROGRAM)

Press -] [-, **2,0,0, ENTER** to select an input.
Press **FUNCT, FUN, ADV, CNTR, ENTER, ENTER**. Press **INTERN**al and select **10** as your number. Press **ENTER**.

Note that the cursor is opposite the second coil of the counter. Press -] [-, **2,0,1, ENTER**. Press the "-" key and hold it down until the "beep" is heard and the setpoint is displayed.

Press --> to position the cursor under contact 2/01.
Press -] [-, **FUNCT, CNTR, 1, ENTER**. Now press -( )-,3,0,0, ENTER.

When input 2/00 is closed, the counter is enabled. You must close input 2/01 ten(10) times to reach the pre-set count. At that count (10) output 3/00 will energize.

```
EXAMPLE:                Rung 001
        2/00                             +--------+
        -] [-----------------------------| Reset  |
        2/01                             |CNTR 1  |
        -] [-----------------------------| 10c    |
                                         +--------+

        CNTR 1                                 3/00
        -] [-----------------------------------( )-
```

6. DR - Drum Sequencer (CLEAR OLD PROGRAM)

Press -] [-, **2,0,0,ENTER** to select an input.
Press **FUNCT, FUN, ADV, DR, ENTER, ENTER.**

The PR-05 is now asking you to program the maximum number of steps (a maximum of 256 can be programmed). Press **5, ENTER**. The PR-05 is now asking you to program the maximum number of channels (a maximum of sixteen can be programmed). Press **5, ENTER**. The PR-05 is now displaying the matrix. Each horizontal row represents a step. Each position in a step (vertical rows) represents a channel. Each channel is a contact and can be used anywhere in the program as required. The state of each contact (on or off) is determined by placing a 1 (on) or a 0 (off) in the position in the Drum matrix and step and stepping the drum as required in the program.

```
EXAMPLE:                      CHANNEL
                              12345
                  S|    1.    00000
                  T|    2.    00000
                  E|    3.    00000
                  P|    4.    00000
                  S|    5.    00000
```

The display also indicates that this matrix is for DR 1. It shows two numbers in brackets.

Press **-] [-, FUNCT, ADV, DR, 1, ENTER.** This identifies the contact as coming from DR 1.

Now identify which channel is to be selected by pressing **1, ENTER.** Now press **-( )- ,2,0,0, ENTER.** Continue these steps for each channel of your drum.

```
EXAMPLE:                    Rung 002
          DR1-1                                2/00
         -] [------------------------------( )-

          DR1-2                                2/01
         -] [------------------------------( )-

          DR1-3                                2/02
         -] [------------------------------( )-

          DR1-4                                2/03
         -] [------------------------------( )-

          DR1-5                                2/04
         -] [------------------------------( )-
```

In running this program, you will find that every time you turn input 2/01 off then back on again, the drum will advance to the next step. If you followed this example correctly, you will see a change in the output LED's.

7. SR - Shift Register (CLEAR OLD PROGRAM)

Press **-] [-,2,0,0, ENTER** to select an input. This will be the reset input.

Press **FUNCT, FUN, ADV, ADV, SR, ENTER, ENTER.** This will program an 8-bit shift register.

Press **-] [-,2,0,1, ENTER.** Now press the "-" key and hold it until a "beep" is heard. Now the display reads CLOCK SR01. You have just programmed the clock input for shift register 1.

Press **-->**. Note that another input is required.
Press **-] [-, 2,0,2, ENTER.** Press and hold the "-" key until a "beep" is heard. Note that the display reads DATA SR01. You have just programmed the data input for SR1.

Press **-->** to position the cursor under input 2/02.
Press **-] [-, FUNCT, ADV, ADV, SR, 1, ENTER.** The PR-05 is asking you to select a channel # press **5, ENTER.** The contact is identified as shift register # 1 channel # 5.
Press **-( )-,3,0,0, ENTER.** This will be the output to energized from bit #5.

When input 2/00 is energized, the shift register is enabled. When input 2/00 is de-energized, SR1 is reset.

Closing input 2/01 will clock the shift register with the status of input 2/02. With input 2/02 closed (ON), toggling 2/01 will load a "1" into the shift register. Continuing to toggle 2/01 will shift the bits one position to the right. When a "1" reaches the 5th bit, the output will energize.

Shift registers are very useful for tracking the position of a part on an assembly line or conveyor. The clock is usually driven by a shaft encoder which gives position displacement.

```
EXAMPLE:                    Rung 001
          2/00                                +--------+
          -] [-------------------------------|Reset   |
                                              | SR 1   |
          2/01                                |        |
          -] [-------------------------------| CLOCK  |
                                              |        |
          2/02                                |        |
          -] [-------------------------------| DATA   |
                                              +--------+

          SR1-5                                    3/00
          -] [------------------------------------( )-
```

To demonstrate how this program works. First turn on input 2/00. Next turn on input 2/02. Now turn input 2/01 on then back off again. Then turn input 2/02 off. If you turn input 2/01 on and off four more times, output 3/00 should energize.

8. ABR - Access Bear Timer/Counter (CLEAR OLD PROGRAM)

The Access Bear is a timer counter access module. All of the setpoints for the timers and counters can be programmed into the Access Bear to allow the operator to have flexibility to change these values. To program the AB-01 as a counter, or retentive timer:

Press -] [-,2,0,0.
Then press **FUNCTION, FUN, ADV, ADV** until ABR appears on the screen.
Press **ABR**
Press **ENTER.**
Next displayed is function #1. Press **ENTER.**

The cursor is now under input 2/00.
Press -] [-,2,0,1, **ENTER**. Now press and hold the "-" key until the "beep" is heard. Now press
"-->" to position the cursor under input 2/01.
Press -] [-, then the blue **FUNCTION** key, **ADV** until **ABR** appears on the screen. The display will show ABR #5. Press **1**, then press **ENTER**. You have now programmed a contact from ABR-1.
Press -( )- ,3,0,0.

When the AB-01 channel #1 reaches its setpoint, output 3/00 will be energized.

NOTE: In order to run this program you must have an Access Bear module connected to your controller

NOTE: TO FIND OUT MORE ABOUT THE AB-01, REFER TO DATA SHEET #7809-48.

```
EXAMPLE:               Rung 001
         2/00                        +--------+
         -] [------------------------|Reset   |
                                     |ABR 1   |
         2/01                        |        |
         -] [------------------------| Inc    |
                                     +--------+

         ABR-1                          3/00
         -] [-----------------------------( )-
```

The above program can be a counter or retentive timer. To modify this to a Time Delay Pick Up, make the following change:

```
EXAMPLE:               Rung 001
         2/00                        +--------+
         -] [-+----------------------|Reset   |
              |                      |ABR 1   |
              |                      |        |
              +----------------------| Inc    |
                                     +--------+

         ABR-1                          3/00
         -] [-----------------------------( )-
```

The Access Bear can be programmed as a Time Delay Dropout. To do this, enter in the following program.

```
EXAMPLE:               Rung 001
          2/00                           CR-2
         -] [-----------------------------(L)-

          CR-2                           3/00
         -] [-+---------------------------( )-
              |
              |                      +--------+
              |----------------------|Reset   |
              |                      |ABR 1   |
              |                      |        |
              +----------------------| Inc    |
                                     +--------+

         ABR-1                           CR-2
         -] [-----------------------------(U)-
```

9. To Change any of the Setpoints (SetPt)

In the previous function examples, simply place the cursor over the lower box in the function so that the setpoint is displayed on the screen. Press the Blue function key. Now press advance until Setpt appears on the display. Press the F1 key and proceed as before in setting up the setpoints.

VII. Programming Instructions

The PR-05 allows direct access to some of the instruction set of the Motorola 14500 processor. Other of these instructions are created by the PR-05 for you. They all provide powerful tools capable of enhancing your program. However, you must use them with care.

1. Latch -(L)- and Unlatch -(U)- (CLEAR OLD PROGRAM)

Press -] [-,2,0,0, **ENTER** to program the input.
Press **FUNCT, INSTR,-(L)-,3,0,0, ENTER**. The latch coil is programmed.

Press -] [-,2,0,1, **ENTER** to program the input.
Press **FUNCT, INSTR,-(U)-,3,0,0, ENTER.**

```
EXAMPLE:                      Rung 001
              2/00                                      3/00
             -] [-----------------------------------(L)-

              2/01                                      3/00
             -] [-----------------------------------(U)-
```

When input 2/00 is closed, output 3/00 is energized. If 2/00 is now opened, output 3/00 stays energized since it is a latched coil. When input 2/01 is closed, output 3/00 is de-energized (unlatched).

Note: There is a problem with this program. If both inputs are closed, output 2/00 will latch and immediately unlatch.

Place the cursor over input 2/00.
Press -->.
Press -]/[-,2,0,1, **ENTER.**
Press ↓, <-- , -]/[-,2,0,0, **ENTER.** The circuit is now interlocked so that one and only one input can be on to latch or unlatch output 3/00.

Press ENTER to go to Rung 2.

```
EXAMPLE:                     Rung 001
          2/00    2/01                         3/00
         -] [----]/[----------------------(L)-

          2/01    2/00                         3/00
         -] [----]/[----------------------(U)-
```

You may now run this program.

2. MCR - Master Control Relay (CLEAR OLD PROGRAM)

Press -] [-,2,0,0, ENTER.
Press **FUNCT, INSTR,** MCR.

The master control relay function is now programmed as a function of input 2/00. If input 2/00 is closed, the MCR will allow all following programs to be executed. If input 2/00 is de-energized, all outputs following the MCR will be turned off.

```
EXAMPLE:                     Rung 001
          2/00
         -] [---------------------------(MCR)-

          2/01                           3/00
         -] [-----------------------------( )-

         -------------------------------(MCR)-
```

After programming a conditional MCR followed by your logic, you should program another unconditional MCR. With the cursor positioned at the left of the display:

Press **FUNCT, INSTR,** MCR. This is an unconditional MCR. No logic is required to enable it. The first MCR instruction is conditional on the state of input 2/00.

Press ENTER to go to Rung 2.
You may now run this program.

Notice that when input 2/00 is off, output 3/01 is de-energized and cannot be energized.

3. OEN - Output Enable (CLEAR OLD PROGRAM)

Press -] [-,2,0,0, ENTER.
Press and hold the "-" key until a "beep" is heard.
Press **FUNCT, INSTR,** ADV, OEN.

The output enable function is now programmed as a function of input 2/00. If input 2/00 is closed, the OEN will allow all following outputs to change state. If the OEN is not enabled, the following outputs cannot change state.

Press -] [-,2,0,1, ENTER.
Press -( )-,3,0,0, ENTER. The internal logic is programmed.

After programming a conditional OEN followed by your logic, you should program another unconditional OEN. With the cursor positioned at the left of the display:

Press and hold the "-" key until a "beep" is heard.
Press **FUNCT, INSTR, OEN.** This is an unconditional OEN. No logic is required to enable it. The first OEN instruction is conditional on the state of 2/00.

```
EXAMPLE:                    Rung 001
           2/00
          -] [---------------------------(OEN)-

           2/01                         3/00
          -] [------------------------------( )-

          -------------------------------(OEN)-
```

The difference between these last two instructions is that the MCR writes all of its outputs OFF when disabled and the OEN leaves the state of its outputs unchanged when disabled. This is sometimes useful to allow a bumpless transfer between one part of the program and the next.

Press ENTER to go to Rung 2.

You may now run this program.

4. CMP - Compare (CLEAR OLD PROGRAM)

Press -] [-,2,0,0, ENTER.
Press **FUNCT, INSTR, ADV, CMP,2,0,1, ENTER.**

Press -( )-,3,0,0, ENTER.

```
EXAMPLE:                    Rung 001
           2/00  2/01                   3/00
          -] [---]CMP[----------------------( )-
```

This instruction allows you to compare the state of two contacts. If both 2/00 and 2/01 are in the same state, either open or closed, true or false, the compare will output a true and output 3/00 will energize. You may program the first input normally closed. This will have the effect of complementing the compare. That is, the CMP will be true then 2/00 and 2/01 are in different states.

Truth Table for CMP in Rung 1.

2/00	2/01	3/00
-] [-	-] [-	-( )-
1	1	1
1	0	0
0	1	0
0	0	1

By changing input 2/00 to a normally closed switch, the truth table becomes:

2/00	2/01	3/00
-]/[-	-] [-	-( )-
1	1	0
1	0	1
0	1	1
0	0	0

```
EXAMPLE:                 Rung 001
          2/00  2/01                          3/00
         -]/[---]CMP[--------------------( )-
```

5. SKZ - Skip (CLEAR OLD PROGRAM)

Press -] [-, 2,0,0, ENTER.
Press FUNCT, INSTR, ADV, SKZ.
Press -( )-,3,0,0, ENTER.

When input 2/00 is energized, output 3/00 will turn on. When input 2/00 is de-energized, output 3/00 remains on. The skip instruction allows you to skip the next instruction in the program if the previous logic is false. Since this functions is just like a latch, most programmers use the latch instruction. Note that if you have a program that used the -]/[- coil, the SKZ will function like an unlatch.

```
EXAMPLE:                 Rung 001
          2/00                                3/00
         -] [--]SKZ[---------------------( )-
```

When you press ENTER to go to the next rung, the PR-05 will compile this SKZ program and convert it to a latch coil.

Press ENTER to go to Rung 2.

```
EXAMPLE:                 Rung 001
          2/00                                3/00
         -] [----------------------------(L)-
```

6. END - End (CLEAR OLD PROGRAM)

Press -] [-,2,0,0, ENTER.
Press **FUNCT, INSTR, ADV, ADV, END.**
Press -] [-,2,0,1, ENTER.
Press -( )-,3,0,0, ENTER.

When input 2/00 is off, you may open and close input 2/01 to energize output 3/00. If you close input 2/00, input 2/01 is no longer able to control output 3/00. Note that this appears to function much like the OEN instruction. The END instruction has a much different purpose. The program stops its scan when it encounters the END instruction and returns to the top of the program.

```
EXAMPLE:                 Rung 001
           2/00
          -] [---------------------------(END)-

           2/01                             3/00
          -] [------------------------------( )-
```

VIII. USING EXTERNAL SETPOINTS

Sometimes, it may be necessary to change a setpoint without having to rewrite the program and burn another EPROM. This can be accomplished by using external setpoints. External setpoints can also be used for any of the setpoints on timers, counters, pulse generators, etc. instead of using the pre-defined software setpoint.

Press **-] [-, 2,1,5**. This will assign our reset line at input 2/15.
Press **FUNCT,FUN,TDPU,ENTER,ENTER,ADV** choose 1m as the time base.
Press **EXTERN**al. This will tell the PR-05 that our setpoint will be chosen externally.
Press **CR, 1, ENTER**. This will assign an external setpoint from CR-001.

By choosing CR-001 as the external setpoint, this means that the next 7 control relays will be assigned to the setpoint. The control relays will correspond to a BCD type fashion in the following order:

CR-001 = 1
CR-002 = 2
CR-003 = 4
CR-004 = 8
CR-005 = 10
CR-006 = 20
CR-007 = 40
CR-008 = 80

```
2/15                            +-------+
-] [----------------------------|Reset  |
                                |TDPU-1 |
                                |CR-001 |
                                | XX m  |
                                +-------+

2/00                              CR-001
-] [-----------------------------( )-
2/01                              CR-002
-] [-----------------------------( )-
2/02                              CR-003
-] [-----------------------------( )-
2/03                              CR-004
-] [-----------------------------( )-
2/04                              CR-005
-] [-----------------------------( )-
2/05                              CR-006
-] [-----------------------------( )-
2/06                              CR-007
-] [-----------------------------( )-
2/07                              CR-008
-] [-----------------------------( )-
```

By energizing certain control relays, the correct setpoint can be obtained.
See following example.

EXAMPLE: For TDPU-01 to have a setpoint of 35m, energize:
CR-001 = 1 or 2/00
CR-003 = 4 or 2/02
CR-005 = 10 or 2/04
CR-006 = 20 or 2/05

IX. RETENTIVE FEATURE - Version 1.80 and above only

The High Density series CPU (ICM-HDCPU-02) has a retentive feature that remembers the state of the contacts used in the program. This is very helpful in the event of power loss. If contacts are programmed with the retentive feature, power can be interrupted without resetting any of the contacts. **Remember:** Only the state of the contacts are retained - not the setpoints. All setpoints will be reset when power gets interrupted.

Note: To access the RETENTIVE feature, the correct CPU type needs to be chosen. To choose correct PLC type:

At the Main Menu:
Press **ADV, Config, PLC, ADV, ADV, rHD-16K.** This will allow you access to the retentive feature while programming.

In the program mode:

Press -] [-, 2,0,0. This is the input to be programmed.
Press -( )-, CR, 1, **FUNCT, YES**. This will make CR-001 a retentive CR - rCR-001
Press -] [-, CR, 1, **ENTER**. Any contact using CR-001 will automatically be retained.
Press -( )-, 3,0,0.

```
2/00                           rCR-001
-[ ]-------------------------( )-

rCR-001                           3/00
-[ ]-------------------------( )-
```

The retentive feature can be used for contacts on timers, counters, control relays, etc. This example is for demonstration purposes only. To fully demonstrate the retentive feature, the program would have to be written to an EPROM. Then the power could be interrupted.

If a retentive contact is to be changed from retentive to non-retentive, or vice versa, use the following procedure:

Position cursor over contact to change.
Press **FUNCT, ADV, ADV, ADV, Retain, Yes or No**

X. TITLE CONFIGURATION (CONFIG)

Now that we have covered all of the programming functions, we need to step through the title configuration. This gives the person writing the program the ability to protect what they have written by using passwords as well as giving the program a title.

First you must press Menu Return to get back to the Main Menu.
Press ADV until the word CONFIG appears on the display.

EXAMPLE: Main Menu

PROM Diag Config

Now press **Config.**

Display: Config Options

Title PLC Port

Now press **Title.**

The words Enter Prog Title will appear on the screen. At this point you may enter your program title of up to 30 characters and press **ENTER**. These can be letters (upper or lower case), or numbers. If no title is needed, press **ENTER**.

The PR-05 will ask you to Enter Revision Number: this number can be up to six (6) characters long.

```
Display:            Enter Revision
                       Number:
```

You may enter a revision number.
Press **ENTER**.

The PR-05 is asking you to enter a date this also can be up to six characters such as MMDDYY.

Press **ENTER**.

The PR-05 is asking for a an Operator Password: this is the lowest level password available. The password used here says the operator can run the program but cannot change anything.

```
EXAMPLE:            Enter Passwords
                       Oper:
```

Press **ENTER**. The PR-05 is now asking for the User Password: This is the next level. This allows the user to change setpoints on counters and timers, run the program, and add their own logic to the end of the program.

```
EXAMPLE:            Enter Passwords
          Oper:
          User:
```

Press **ENTER**. Now the PR-05 is asking for the OEM password: This is the master password and will allow you to change anything in your program.

```
EXAMPLE:            Enter Passwords
          Oper:
          User:
          OEM:
```

Press **ENTER**.

Any or all of this title information can be skipped over by just pressing ENTER when the one you wish to skip is displayed. All of these passwords can contain up to six characters: letters (Upper or Lower case) or numbers.

Caution

****

We would like to caution you that if you forget the OEM password you must re-write the program. Unless you take the time to set up a good filing system, do not make your passwords too complex.

****

This concludes this tutorial seminar. If you have any comments or would like to see something added to this seminar please send your comments to:

Divelbiss Corporation
9776 Mount Gilead Road
Fredericktown, Ohio 43019

If you have any questions, feel free to call Divelbiss Corporation at 800-245-2327.

CONTROL RELAY MAP

Free Control Relays (CR-001 to CR-107)

Function #1 (CR-108 to CR-119)

Function #2 (CR-120 to CR-131)

Function #3 (CR-132 to CR-143)

Function #4 (CR-144 to CR-155)

Function #5 (CR-156 to CR-167)

Function #6 (CR-168 to CR-179)

Function #7 (CR-180 to CR-191)

Function #8 (CR-192 to CR-203)

Function #9 (CR-204 to CR-215)

Function #10 (CR-216 to CR-227)

Function #11 (CR-228 to CR-239)

Function #12 (CR-240 to CR-251)

Function #13 (CR-252 to CR-263)

Function #14 (CR-264 to CR-275

Function #15 (CR-276 to CR-287)

Function #16 (CR-288 to CR-299)

Function #17 (CR-300 to CR-311)

Function #18 (CR-312 to CR-323)

Function #19 (CR-324 to CR-335)

Function #20 (CR-336 to CR-347)

Function #21 (CR-348 to CR-359)

Function #22 (CR-360 to CR-371)

Function #23 (CR-372 to CR-383)

Function #24 (CR-384 to CR-395)

Function #25 (CR-396 to CR-407)

Function #26 (CR-408 to CR-419)

Function #27 (CR-420 to CR-431)

Function #28 (CR-432 to CR-443)

Function #29 (CR-444 to CR-455)

Function #30 (CR-456 to CR-467)

Function #31 (CR-468 to CR-479)

Function #32 (CR-480 to CR-491)

NOTES