

## MDI3x00 Special Multi-label Decoding

### Purpose:

This application note discusses several new commands to configure the MDI3x00 family of modules to decode multiple bar code symbols in one trigger cycle. This document is based on firmware #BA01J08 or newer. There are other “Read Modes” of decoding the same, or multiple barcodes (See section 5.1 in the “Universal Menu Book”), but for the purpose of this document we will discuss, the newest modes and commands.

The sections below discuss several configuration commands that can assist in decoding and outputting multiple labels in a single read cycle sequence.

### 1. Multi-Label reading: [D3P]

This command is the main configuration code that now instructs the imager to search and decode for as many unique symbols as possible while in a triggered read cycle. Once a symbol has been decoded, the bar code data is tested to see if it is unique (had it been decoded before in this read cycle?). If it is a unique bar code, then it is stored in the memory list and it is immediately outputted, a good read beep tone is issued (unless disabled) and the module remains in a triggered read cycle until the read cycle is canceled or the read cycle timer has timed out.

Changing the read mode configuration to any one of the following will cancel the one mode and change to the new read mode specified (they are mutually exclusive):

Command	Name	Function
S0	Single Read Mode	After a single decode, the read cycle terminates
S1	Multiple Read Mode	In this mode, the reader will attempt to decode any symbol and remain in the Read Cycle when it can no longer successfully decode for a time set by the Read Time option settings or until the read time is canceled. A unique symbol is decoded once and the same symbol will not be decoded again in the same read cycle unless the same label has not been detected (removed from view) for a set number of frames. Each unique decode and output will reset the Read Cycle timeout timer making it possible to decode as long as new symbols are presented.
S2	Continuous Read Mode	The reader will read and output as fast as possible any symbol it can decode including the same symbol for the set read cycle time. This mode is mainly used for diagnostics.
D3P	Multi-Label Mode	This mode is similar in scope to the “Multiple Read Mode” (S1), however it reads and outputs a unique label only once during a read cycle. Also, the read cycle time-out timer is not reset on a good decode., so this mode will decode all unique symbols for a set period of time.

## 2. Output on a decoding failure (Output “BAD\_READ” message): [D8H

The MDI3x00 firmware has a special output message when a symbol does not decode properly. The output format looks like: "**BAD\_READ (x = nnn, y = mmm)!**". This setting must also be used in conjunction with the CENTER coordinate output mode for decoded data (See section 5).

For example:

If a symbol can be located but for some reason cannot be fully decoded, then the result is to obtain center point of the symbol, output the "BAD\_READ (x = nnn, y = mmm)!" message (where nnn & mmm are the center pixel location) and the image is tossed to take a new image and attempt another decode.

This BAD\_READ message is considered a unique barcode if the (X=nnn, Y=mmm) numbers are at all different (even if off by 1). So for instance, if a label of say six symbols is presented to the reader and the [D8H mode is set. Then if all symbols decoded correctly the first time, then there will be an output of six decoded labels, however, if during the image captures, one symbol is located, but does not decode, then the BAD\_READ message will output and a new capture, locate and decode will be attempted. If the symbol that did not decode before, now does decode, then the output will show that decoded data, since this output is unique from the BAD\_READ data. In this scenario, you would then see 7 outputs (6 of decoded data and 1 BAD\_READ message).

To verify that a symbol is truly a BAD\_READ, and had not later decoded, perform this test: At the end of the read cycle, compare the BAD\_READ's CENTER pixel location with the X,Y pixel location output of all the decoded data (command: MZ[DDY in section 5). Be sure to allow for some tolerance of pixel shift (i.e. +/- 10 pixels). If the BAD\_READ location does not match the location of a good decode location, then the BAD\_READ is valid, otherwise if it did match a good read location, then the BAD\_READ must be discarded.

Note: There is the possibility that the module fails to decode the symbol when trying to determine between two adjacent symbols (trying to determine where one ends and the other begins). This determination may render an output as a single BAD\_READ message with the center coordinates being the midpoint between the two separate symbols. (i.e. depending on the failure mode, the positional data may not be accurate to a specific 'single' symbol).

*\*\*:* This "BAD\_READ" output specification behaves differently from the MDI-1000 firmware family.

## 3. Fast search algorithm: [DE9

For well printed symbols (black on white, good quality), it is recommended that you use this "Fast search algorithm". This algorithm speeds up the locate time (less time spent to locate). Thus, it is possible to increase the probability of decoding all symbols in a small amount of time.

## 4. Reading one Symbology only: (Data Matrix: [BC0) / (QR Code: [BC1)

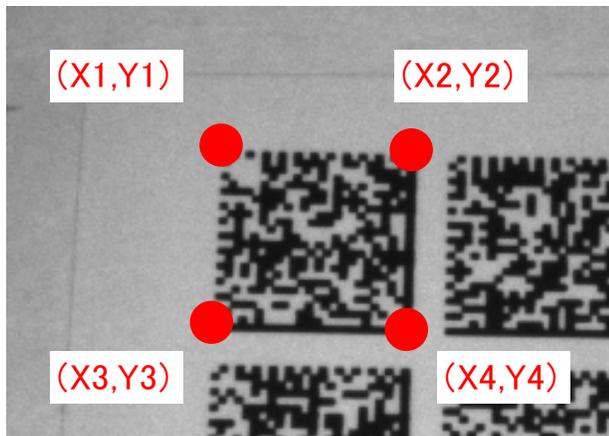
It is recommended to turn off all symbologies, other than the symbology you are decoding. This eliminates the symbology type searching and allows for a quicker decoding process which improves the reading performance. For reading Data Matrix symbologies only use the command: [BC0. For reading QR symbologies only, use the command: [BC1. If there is a need to read either symbology, then send the command: [BC0 first to set Data Matrix only, and then send the command: [BCD to add the QR code as an additional symbology to check (this will not be as fast as only one symbology enabled).

## 5. Send Coordinate output: (Rectangle: MZ [DDX) / (Center: MZ [DDY)

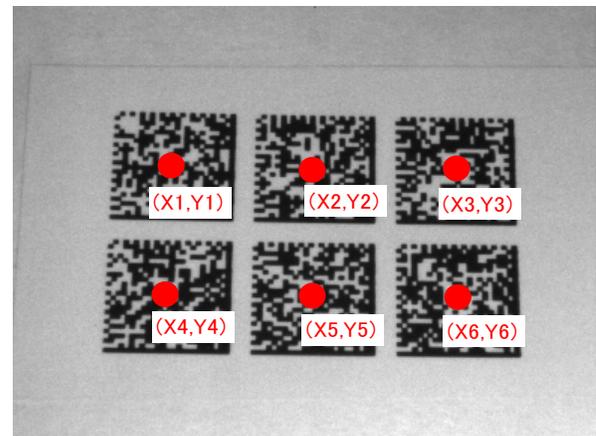
This mode allows the data output to be sent with the pixel location of where the symbol was found. There are two output formats that can be set:

- Rectangle Format Output
- Center Format Output
- Normal

It may be necessary to know the positional data of the symbol that had been decoded. Using positional data, it may then be determinable if any symbols in a specific location were decoded or not.



"Rectangle" format



"Center" format

### To set the Coordinate output to "Rectangle" format:

Send the command in one command string: **MZ [DDX**

The decoded data will now output in the following format:

Format : **X1,Y1;X2,Y2;X3,Y3;X4,Y4;{decode\_data}**

Example : **+0532,+0287;+0334,+0285;+0334,+0247;+0532,+0251;{decode\_data}**

### To set the Coordinate output to "Center Point" format:

Send the command in one command string: **MZ [DDY**

The decoded data will now output in the following format:

Format : **X1,Y1; {decode\_data}**

Example : **+0532,+0287; {decode\_data}**

*Note: This center mode will also output the X,Y coordinates before the BAD\_READ message. The center coordinates will always be the same as listed within the BAD\_READ (coordinates).*

**+0446,+0280;BAD\_READ(x=446,y=280)!<CR>**

### To clear the Coordinate output to a Normal format:

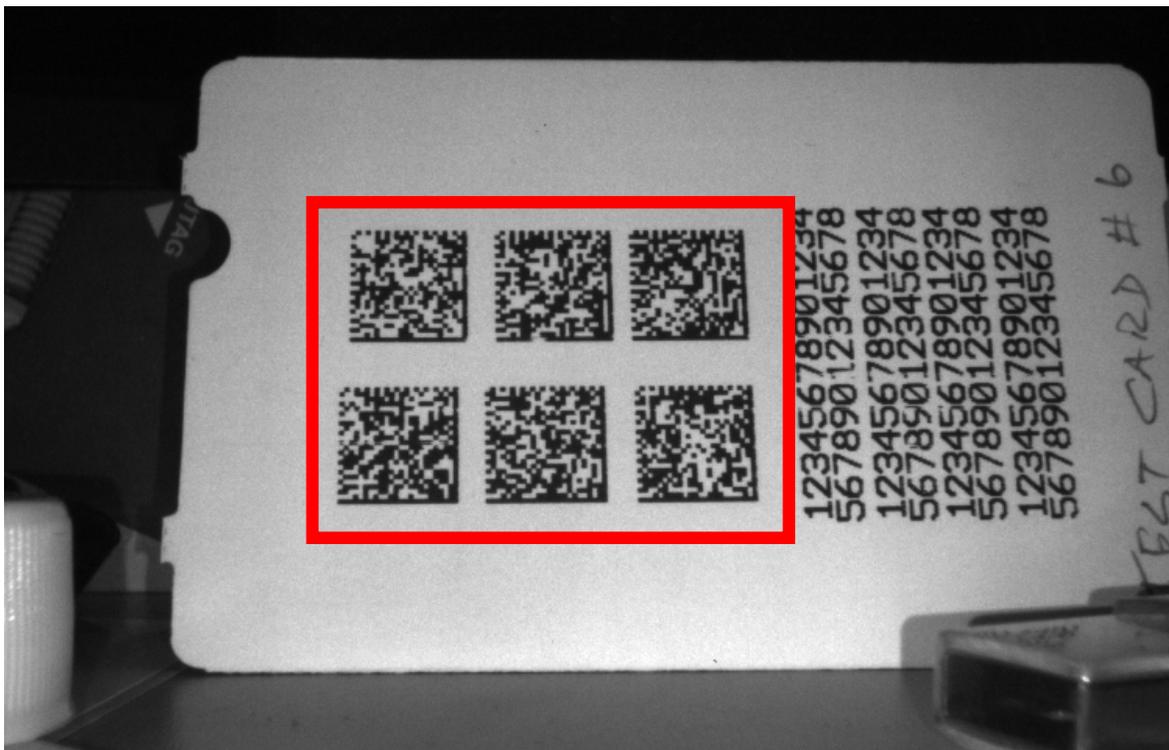
Send the single command string: **MZ**

## 6. Specifying a Region of Interest: [DF8]

It is possible to specify a rectangular search area region within the image captured. This allows the bar code reader's 'locate process' to focus on only looking in a specific area of the image to find and attempt a decode of any particular symbol.

For example, human readable text will typically confuse the reader into thinking there is a symbol located here. This will then take up some decode time trying to sort it out, but it will eventually drop it. If it knew it did not look like a symbol, then it immediately goes back to the locate process and looks for another candidate. If it thought the found text looked enough like a symbol, but could not decode it, it would output a "BAD READ" message.

To avoid spending the unnecessary time in any known garbage area of the image, the answer is to specify a region of interest (ROI) to avoid such things. This will also have the added benefit of speeding up the decoding process.



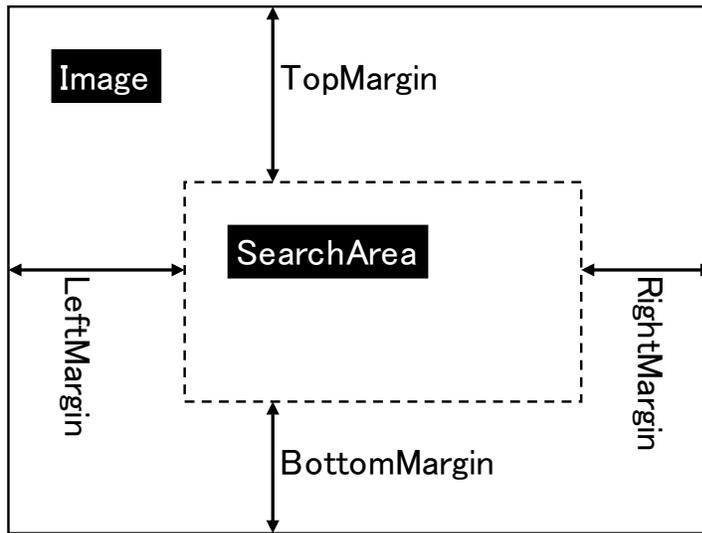
*The orange outline specifies a region of interest*

The region of interest specifies the region of the image to search for a symbol. So if a symbol hangs partially inside and also outside of the region of interest that has been set, it will be enough to locate the code and then the decode algorithm will work on the whole symbol. (i.e. Decoding does not require that the entire symbol be contained within the Search-Area, only a portion of it needs to be inside to be located).

**To set the Region of Interest (Search Area):**

After the command ID: [DF8, it is to be followed by sending numeric values (direct input numbers, where: Q0=0 to Q9 = 9) that represents the register to modify and the value that represents the 4-digit margin (in # of pixels).

By specifying the margins: Top-Margin / Right-Margin / Bottom-Margin / Left-Margin like shown in the drawing on the next page, it will subtract these from the sensor size (i.e. MDI3100 has a sensor size of 752 x 480) and it will create the Search-Area in the Image. For instance A margin of Q0=100, Q1=100, Q2 = 100, & Q3 = 100 will produce a region in the center of the sensor that is 552 x 280 pixels in size. Issuing the command: <ESC>[DF8Q4Q0Q0Q0Q0<CR> will clear all margins and the search area will be the whole sensor size.



	ID	1digit	2~5digit
Top Margin (max 190)	[DF8	Q0	Q0Q0Q0Q0 ~ Q9Q9Q9Q9 (*)
Right Margin (max 326)		Q1	
Bottom Margin (Max 190)		Q2	
Left Margin (max 326)		Q3	
Set Default		Q4	

(\*) The range of values that can be set depends on the model. If an out-of-range value is entered, the number closest to the number entered in the range of the number is set. Also: There are specific ranges that cannot be set, such as in the center of the image. These areas will be excluded from creating a boundary as the value of the margin may be too large.

## 7. Typical Configuring for Multi-Label Reading

To get a quick setup going, scan the configuration code below

*Multi-Label Reading Configuration Example:*



@MENU\_OPTO@ZZ@BC0@D3P@D8H@DE9@MZ@DDY@ZZ@OTPO\_UNEM

The symbol above, configures the reader for:

Data Matrix Only + Enable Multi-Label reading + Enable the “BAD\_READ” output + Enable the Fast search algorithm + Enable the Coordinate output as “Center” format

**You may also want to set a Read Cycle Time out:**

Scan one of the configuration codes below to set a specific read cycle time out period:

2 Seconds	3 Seconds	4 Seconds	5 Seconds
Command: Y2	Command: Y3	Command: Y4	Command: Y5

@MENU\_OPTO@ZZ@Y2@ZZ@OTPO\_UNEM



**Read Cycle = Disabled [Y0]**

(Use this mode to control the cycle through software: ‘Z’ command to start the trigger read cycle & use the ‘Y’ command to end the read cycle.

**Then set a region of interest if you want:**

Format: [DF8Q0QnQnQnQn[DF8Q1QnQnQnQn[DF8Q2QnQnQnQn[DF8Q3QnQnQnQn

Example: [DF8Q0Q0Q0Q6Q0[DF8Q1Q0Q1Q0Q0[DF8Q2Q0Q1Q8Q0[DF8Q3Q0Q2Q2Q5

*Note: The values in this example are suggestive, and may not reflect the actual values required in the specific user application.*

The example command would set a region of interest as:

Top Margin	60 pixels
Right Margin	100 pixels
Bottom Margin	180 pixels
Left Margin	225 pixels



@MENU\_OPTO@ZZ@[DF8Q0Q0Q0Q6Q0@[DF8Q1Q0Q1Q0Q0@[DF8Q2Q0Q1Q8Q0@[DF8Q3Q0Q2Q2Q5@ZZ@OTPO\_UNEM@

