

## Programming Based on A Variable Value Being Odd or Even

### Introduction

Our INFORM language does not have an instruction that can directly determine if the value of a variable is odd or even. PLC programmers using RS Logix can do this and may be looking for a way to do this with INFORM instructions. Below we will introduce an example scenario and the steps needed to accomplish this. NOTE: this will not determine the value of the variable, only if it is an even or an odd number. The purpose of this document is just to show you how to find out if a variable value is even or odd, there are of course other ways to track this programmatically.

### Scenario

We are incrementing Variable I000 with every part that we place on our pallet, and we have an odd shape at one end of the part. We would like to add an offset for every other part to allow us to stack tighter. We would like to track this by checking if I000 is an odd or an even number.

### Solution

To do this, we will need to select an unused output group (a byte) to write to. This example will use Output Group #6. Now you can use the SET command write the value of the integer to the output group.

```
J:Odd-Even  
CONTROL GROUP: R1  
0000 NOP  
0001 SET OG#(6) I000  
0002 END
```

Now that the variable is written to the output group, all we need to do is monitor the first bit of that group, in our case Output#41. If it is on, the value of the variable must be an odd number. If it is off, the value must be an even number. We can monitor this with an if then statement and continue our programming.

This document captures ideas, experiences, and informal recommendations from the Yaskawa Partner Support team. It is meant to augment – not supersede manuals or documentation from motoman.com. Please contact the Partner Support team at [partnersupport@motoman.com](mailto:partnersupport@motoman.com) for updates or clarification.

```
JOB CONTENT: MASTER
J:Odd-Even
CONTROL GROUP: R1
0001 SET OG#(6) I000
0002 '*
0003 'check for odd #
0004 IF( OT#(41)=ON ) THEN
0005 SFTON P000
0006 MOVL P050 V=138
0007 SFTOF
0008 '*
0009 'check for even #
0010 ELSEIF( OT#(41)=OFF ) THEN
0011 MOVL P050 V=138
0012 ENDIF

MOVJ VJ=25.00
```

This works, because the first bit of an odd number is always 1. See below.

Decimal Value	Binary Representation of Decimal Value	Last Bit of the Binary	Odd or Even
0	00000000	0	Even
1	00000001	1	Odd
2	00000010	0	Even
3	00000011	1	Odd
4	00000100	0	Even
5	00000101	1	Odd
6	00000110	0	Even
7	00000111	1	Odd
8	00001000	0	Even
9	00001001	1	Odd
10	00001010	0	Even