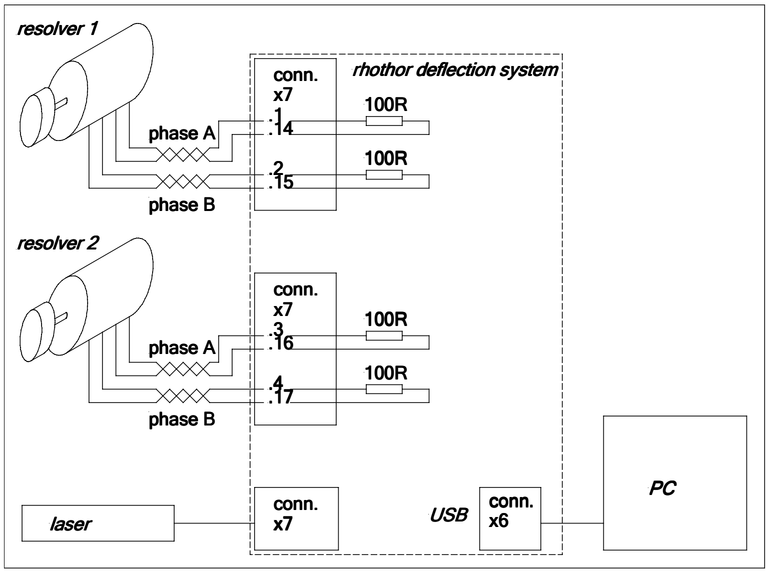
APPLICATION NOTE
MARKING ON THE FLY
FOR "CUA-USB" AND "CUA-ETH" CONTROL UNITS

NEWSON ENGINEERING NV

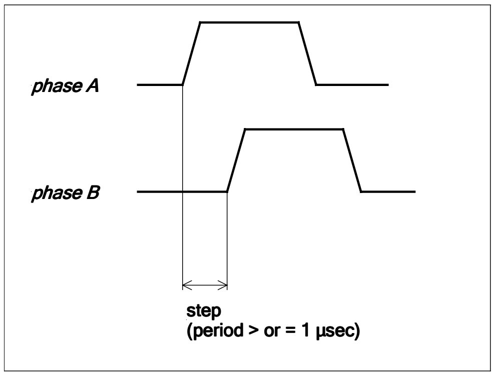
Table of Contents

1 Using Marking-on-the-Fly to Increase Marking Area

When a rhothor™ deflection control system is connected with one or two resolvers, marking areas larger than the system's scanfield are easily implemented. The resolvers are used to measure the position of a movable table. When the rhothor™ deflection system or the workpiece is mounted on that table, the scan area of the deflection system is extended with the travel range of the table.



The resolvers generate 2 phase-shifted signals, phase A and Phase B. Each flank is counted as a step. The size of the step has to be declared to the rhothor™ system.



<u>function</u>: rtSetResolver(nr,stepsize,countrange);

<u>parameters</u>:
nr: 1 or 2
stepsize: size of one step, negative values to invert the movement
countrange: max countvalue of the resolver, has to be set to 0

The function rtSetResolver will not only set the resolvers stepcount but will also reset its internal counter. The table has to be positioned in a reference position before this function is called. This reference position will then serve as the zero position of the marking field. All co-ordinates used in communication over the rhothor.dll are referenced to this table position. This referencing can be done on a daily basis or on a job basis.

Example code:

```
...
rtListOpen(1);
// set stepsize X resolver at 1 µm, reset X resolver counter
rtSetResolver(1,0.001,0);
// set stepsize Y resolver at 0.5 µm, reset Y resolver counter
rtSetResolver(2,0.0005,0);
rtListClose();
// wait until processed
while (rtGetStatus(NULL)==ERR_BUSY);
...
```

After initialising the resolvers job commands can be send to the rhothor™ deflection control system. All co-ordinates used in the command stream are referenced to the table position when the function rtSetResolver was called. When the table moves, the movement will automatically be added to the positions used in the command stream. The range of the actual deflection system however has not changed. If the resolver counts are higher than the fieldsize even a rtJumpTo(0,0) (jump to centre position) will be impossible to execute. On the other hand if the X table has moved to 500mm for example, a rtJumpTo(500,0) will be executable by the system even though it seems well out of range of the deflection system itself.

function: rtWaitResolver(nr,position,mode);

parameters:
nr: 1 or 2
position: co-ordinate
mode: 1(2) resolver counter has to be larger(smaller) than co-ordinate

This function will suspend execution of the command stream until the position is reached by the resolver.

Example code:

```
...
rtListOpen(1); // A
rtWaitresolver(1,500,1); // B
rtJumpTo(500,0); // C
rtLineTo(500,10); //D
rtListClose();
...
```

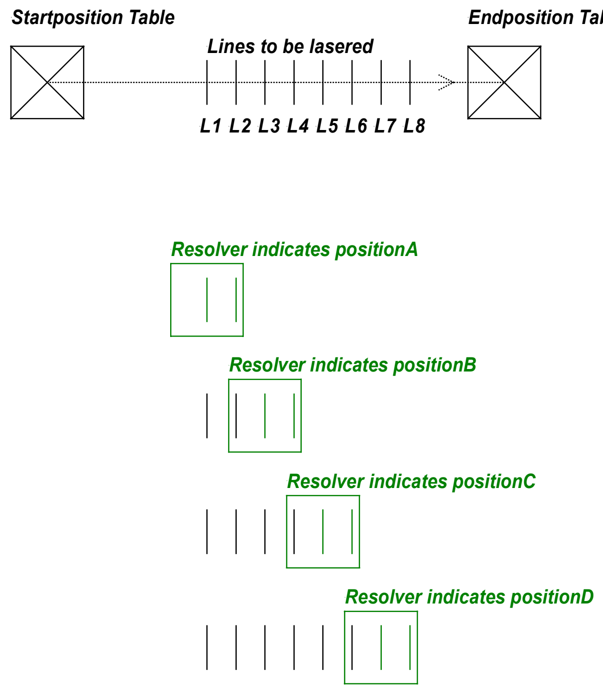
A: At this point of the code, let us assume that the table is moving at a constant speed towards X = 1000 mm
B: This command suspends the execution of following commands. The sending of commands can be continued over the dll. They will all be stored in the FIFO. When the X-axis of the table reaches 500 mm, code execution resumes.
C: This is the first command after the rtWaitResolver command. The X position of the table is 500 mm, so the rtJumpTo(500,0) command is seen by the deflection control system as a jump towards its centre position.
D: rtLineTo(500,0) is seen by the deflection control system as an interpolation from its centre point (0,0) to (0,10).

Limitations:

Speed: Step frequency of the resolver must be lower than 1 MHz
Size: The overall fieldsize (XY table travel) must be lower than 127 times the scanner fieldsize.
Co-ordinates: The co-ordinates during the command stream must be executable by the deflection control system. The

range is therefore limited to a field with centre position at the last co-ordinates used in the rtWaitResolver functions and with a size equal to the fieldsize of the deflection system.

Simple example:


Startposition Table
Endposition Table
Lines to be lasered
L1 L2 L3 L4 L5 L6 L7 L8
Resolver indicates positionA
Resolver indicates positionB
Resolver indicates positionC
Resolver indicates positionD

Command send to rhothor deflection system

```
rtListOpen()
rtWaitResolver(1,A,1)
L1
L2
rtWaitResolver(1,B,1)
L3
L4
rtWaitResolver(1,C,1)
L5
L6
rtWaitResolver(1,D,1)
L7
L8
rtListClose()
```

Command send to table

StartTable (from StartPosition to EndPosition)