

```
//Example linear toolchanger code
```

```
//Tool positions definition
```

```
int Chuckopenport = 1; port on uc 300/400
```

```
int Chuckopenpin = 16; output pin number
```

```
double[] ToolX = new double[11];
```

```
double[] ToolY = new double[11];
```

```
ToolX[0] = 0; // Tool0 X position
```

```
ToolY[0] = 0; // Tool0 Y position
```

```
ToolX[1] = 50; // Tool1 X position
```

```
ToolY[1] = 10; // Tool1 Y position
```

Fill in right numbers of

tool positions

```
ToolX[2] = 100; // Tool2 X position
```

```
ToolY[2] = 10; // Tool2 Y position
```

```
ToolX[3] = 150; // Tool3 X position
```

```
ToolY[3] = 10; // Tool3 Y position
```

```
ToolX[4] = 200; // Tool4 X position
```

```
ToolY[4] = 10; // Tool4 Y position
```

```
ToolX[5] = 250; // Tool5 X position
```

```
ToolY[5] = 10; // Tool5 Y position
```

```
ToolX[6] = 300; // Tool6 X position
```

```
ToolY[6] = 10; // Tool6 Y position
```

```
ToolX[7] = 350; // Tool7 X position
```

```
ToolY[7] = 10; // Tool7 Y position
```

```
ToolX[8] = 400; // Tool8 X position
```

```
ToolY[8] = 10; // Tool8 Y position
```

```
ToolX[9] = 450; // Tool9 X position
```

```
ToolY[9] = 10; // Tool9 Y position
```

```
ToolX[10] = 500; // Tool10 X position
```

```
ToolY[10] = 10; // Tool10 Y position
```

```
double SafeZ = 100; mm to retract
```

```
double Ztoolrelease = 30; hight of relaese
```

```
double Ztoolpickup = 28; hight of pick up
```

```
int Newtool = exec.Getnewtool();
```

```
int Currenttool = exec.Getcurrenttool();
```

```
if(Newtool == -1) // If new tool number is -1 means a missing T code, so we need to stop here...
```

```
return; safety
```

```
if(Newtool < 1 || Newtool > 10) // Tool number is out of range, so we need to stop here...
```

```
return; safety
```

```
if(Newtool == Currenttool) // Same tool was selected, so do nothing, stop here...
```

```
return; safety
```

```
if(!exec.GetLED(56)||!exec.GetLED(57)||!exec.GetLED(58)) // If machine was not homed then it is unsafe to  
move in machine coordinates, stop here...
```

```
{
```

```
    MessageBox.Show("The machine was not yet homed, do homeing before executing a tool change!");
```

```
exec.Stop();  
return; safety  
}
```

```
while(exec.IsMoving()){}
```

```
// Get current XY machine coordinates to return to this position at the end of the macro
```

```
double Xoriginalpos = exec.GetXmachpos();  
double Yoriginalpos = exec.GetYmachpos();
```

```
// Stop spindle if running and Move Z up
```

```
exec.Stopspin();  
exec.Code("G00 G53 Z"+ SafeZ); // Move Z up  
while(exec.IsMoving()){}
```

```
if(Currenttool!=0) // No need to drop down tool if current tool number is zero
```

```
{
```

```
    // Open toolrack here I have to set port and pin I think  
    exec.Setoutpin ?
```

```
// Move to old tool position on XY plane
```

```
exec.Code("G00 G53 X" + ToolX[Currenttool] + " Y" + ToolY[Currenttool]);  
while(exec.IsMoving()){}
```

```
// Drop current tool
```

```
exec.Code("G00 G53 Z"+ Ztoolrelease); // Move Z axis down to tool holder position  
while(exec.IsMoving()){}  
exec.Setoutpin(Chuckopenport, Chuckopenpin); // Open the chuck with pneumatic valve  
exec.Wait(1000); // Wait one 1000msec  
exec.Code("G00 G53 Z"+ SafeZ); // Move Z up  
while(exec.IsMoving()){}  
}
```

```
// Move to new tool position on XY plane
```

```
exec.Code("G00 G53 X" + ToolX[Newtool] + " Y" + ToolY[Newtool]);  
while(exec.IsMoving()){}
```

```
// Pick new tool
```

```
exec.Code("G00 G53 Z"+ Ztoolpickup); // Move Z axis down to tool holder position  
while(exec.IsMoving()){}  
exec.ClROUTpin(Chuckopenport, Chuckopenpin); // Close the chuck with pneumatic valve  
exec.Wait(1000); // Wait one 1000msec  
exec.Code("G00 G53 Z"+ SafeZ); // Move Z up  
while(exec.IsMoving()){}
```

```
// Close toolrack

// Move back to start point

exec.Code("G00 G53 X" + Xoriginalpos + " Y" + Yoriginalpos);
while(exec.IsMoving()){

// Measure new tool will go here.... Coordinates X Y of fixed toolplate
//exec.Code("G43 H"+Newtool); // Load new tool offset

exec.Wait(200);
while(exec.IsMoving()){
if(!exec.IsMacroStopped()) // If tool change was not interrupted with a stop only then validate new tool number
{
    exec.Setcurrenttool(Newtool); //Set the current tool -> the new tool
    MessageBox.Show("Tool change done.");
}
else
{
    exec.StopWithDeccel();
    MessageBox.Show("Tool change was interrupted by user!");
}
}
```