```
//Example linear toolchanger code

//Tool positions definition
int Chuckopenport = 1;
int Chuckopenpin = 16;

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
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;
double Ztoolrelease = 30;
double Ztoolpickup = 28;

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;

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

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

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;
}

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()){}

//Slide OUT for double acting solenoid

exec.Setoutpin ( 4,10 );    ( energize OUT solenoid  port 4,pin 10 example)
exec.Wait(100);
while (AS3.GetLED(200) == false)  //Wait for input signal for OUT
{
exec.Wait(10);
if(exec.Ismacrostopped()){return;}
}
exec.AddStatusmessage(" Slide is OUT ");

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

//Slide IN For double acting solenoid
exec.Clroutpin ( 4,10 );    ( Denergize OUT solenoid  port 4,pin 10 example)
exec.Wait(100);
while (AS3.GetLED(201) == false)  //Wait for input signal for IN
{
exec.Wait(10);
if(exec.Ismacrostopped()){return;}
}
exec.AddStatusmessage(" Slide is IN ");

// Move back to start point

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

// Measure new tool will go here....
//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!");
}
```