

```

// =====
// UCCNC M6 macro - 12 állásos esernytár (B tengely)
// + 4 bites abszolút tárkód VALIDÁLÁS LED-ekről (nincs korrekció → hiba és Stop)
// + T0 → szerszám lerakás (drop only, nincs felvétel)
// =====

// ----- PARAMÉTEREK / I/O (TÖLTSD KI A SAJÁT GÉPEDHEZ) -----
const int MaxToolNum = 12;          // 12 fészek (T1..T12)
const double ToolBOffsetDeg = 0.0; // ha T1 nem pont B=0°, itt tudsz eltolni (+/- fok)

int ToolUnclampPort = 2;           // NYIT (UNCLAMP) port
int ToolUnclampPin = 14;           // NYIT (UNCLAMP) pin
// ZÁR: ugyanennek a kimenetnek a kikapcsolása (Clroutpin). Ha nálad külön kimenet a ZÁR, azt
// add meg külön és használd azt.

// (opcionális) porelszívó
int DustCollectorPort = 2;
int DustCollectorPin = 1;

// 4 bites abszolút tárkód - LED kódok (A TE kiosztásod szerint!)
// Ezeket a screensetben beállított User LED kódokra cseréld:
int BIT_LED0 = 1001; // LSB
int BIT_LED1 = 1002;
int BIT_LED2 = 1003;
int BIT_LED3 = 1004; // MSB

// ----- Gépi koordináták (G53) - ellenőrizd! -----
double SafeZ = -0.5;
double ToolDown = -5.339;
double ToolUp = 0.0;
double YToolChangePos = -7.8;
double YToolDockPos = -11.6913;

// ----- B fészekszögek (30°/fészek 12 állásnál) -----
double[] ToolB = new double[MaxToolNum + 1];
double bstep = 360.0 / MaxToolNum; // 30°
ToolB[0] = 0.0;
for (int i = 1; i <= MaxToolNum; i++)
{
    ToolB[i] = ToolBOffsetDeg + bstep * (i - 1);
}

// ----- Állapotlekérdezés -----
int Newtool = exec.Getnewtool(); // Tn
int Currenttool = exec.Getcurrenttool(); // tokmányban lévő

// Ha M6 T nélkül → kilép
if (Newtool == -1) return;

// HOME LED-ek (X/Y/Z/B): 56..(legalább B=60) → legyenek aktívak
if (!AS3.GetLED(56) || !AS3.GetLED(57) || !AS3.GetLED(58) || !AS3.GetLED(60))
{
    MessageBox.Show("A gép nincs homingolva. Végezz HOME-ot, majd M6!");
    exec.Stop();
    return;
}

// T0 → csak lerakás
bool onlyDrop = (Newtool == 0);

// Ha nem T0: csak 1..12 engedett
if (!onlyDrop && (Newtool < 1 || Newtool > MaxToolNum))
{
    MessageBox.Show("The TOOL # must be 1 - " + MaxToolNum + " (vagy T0 a kirakáshoz).");
    exec.Stop();
    return;
}

// Ha ugyanaz a szerszám és nem T0 → nincs teendő

```

```

if (!onlyDrop && Newtool == Currenttool) return;

// ----- Előkészítés: orsó leáll, Z fel -----
double Xoriginalpos = exec.GetXmachpos();
double Yoriginalpos = exec.GetYmachpos();

exec.Code("G00 G53 Z" + SafeZ);
while (exec.IsMoving()) {}

exec.Code("M5");
while (exec.IsMoving()) {}

double SpindleSpeed = 0.0;
do
{
    exec.Wait(50);
    SpindleSpeed = Convert.ToDouble(AS3.Getfield(870)); // Actspindlespeed
}
while (SpindleSpeed > 0.1);

// (opcionális) porelszívó bekapcs
exec.Setoutpin(DustCollectorPort, DustCollectorPin);

// =====
// 1) LEDOBÁS (ha van a tokmányban valami)
// =====
if (Currenttool != 0)
{
    // Forgás az AKTUÁLIS fészek fölé
    exec.Code("G00 G53 B" + ToolB[Currenttool]);
    while (exec.IsMoving()) {}

    // 4 bites kód VALIDÁLÁS (korrekció nincs, hiba és Stop)
    int codeA = 0;
    if (AS3.GetLED(BIT_LED0)) codeA |= 1;
    if (AS3.GetLED(BIT_LED1)) codeA |= 2;
    if (AS3.GetLED(BIT_LED2)) codeA |= 4;
    if (AS3.GetLED(BIT_LED3)) codeA |= 8;
    int expA = Currenttool; // 1..12 (a kiosztásod szerint 0001..1100)
    if (codeA != expA)
    {
        string expbinA = Convert.ToString(expA, 2).PadLeft(4, '0');
        string readbinA = Convert.ToString(codeA, 2).PadLeft(4, '0');
        MessageBox.Show("Carousel code mismatch at DROP!
Várt fészek: T" + expA + " (" + expbinA + ")
Olvasott: " + readbinA);
        exec.Stop();
        return;
    }

    // Megközelítés és drop
    exec.Code("G00 G53 Y" + YToolChangePos); while (exec.IsMoving()) {}
    exec.Code("G00 G53 Z" + ToolDown); while (exec.IsMoving()) {}
    exec.Code("G01 F30 G53 Y" + YToolDockPos); while (exec.IsMoving()) {}
    exec.Code("G04 P0.5"); while (exec.IsMoving()) {}

    exec.Setoutpin(ToolUnclampPort, ToolUnclampPin); // UNCLAMP (nyit)
    exec.Code("G04 P0.5"); while (exec.IsMoving()) {}

    exec.Code("G01 F10 G53 Z" + (ToolDown + 0.10)); while (exec.IsMoving()) {}
    exec.ClROUTpin(ToolUnclampPort, ToolUnclampPin); // CLAMP (üres)
    exec.Code("G04 P0.2"); while (exec.IsMoving()) {}

    exec.Code("G00 G53 Z" + SafeZ); while (exec.IsMoving()) {}
    exec.Code("G00 G53 Y" + YToolChangePos); while (exec.IsMoving()) {}
}

// Ha T0 volt a kérés → visszatérés és kilépés (nincs felvétel)
if (onlyDrop)

```

```

{
    exec.Code("G00 G53 X" + Xoriginalpos + " Y" + Yoriginalpos); while (exec.IsMoving()) {}
    exec.Code("G49"); exec.Wait(100); while (exec.IsMoving()) {}

    if (!exec.Ismacrostopped())
    {
        exec.Setcurrenttool(0); // nincs szerszám a tokmányban
    }
    else
    {
        exec.StopWithDeccel();
        MessageBox.Show("Tool change was interrupted by user!");
    }
    return;
}

// =====
// 2) FELVÉTEL (ÚJ SZERSZÁM)
// =====
exec.Code("G00 G53 B" + ToolB[Newtool]);
while (exec.IsMoving()) {}

int codeB = 0;
if (AS3.GetLED(BIT_LED0)) codeB |= 1;
if (AS3.GetLED(BIT_LED1)) codeB |= 2;
if (AS3.GetLED(BIT_LED2)) codeB |= 4;
if (AS3.GetLED(BIT_LED3)) codeB |= 8;
int expB = Newtool;
if (codeB != expB)
{
    string expbinB = Convert.ToString(expB, 2).PadLeft(4, '0');
    string readbinB = Convert.ToString(codeB, 2).PadLeft(4, '0');
    MessageBox.Show("Carousel code mismatch at PICKUP!");
    Várt fészkek: T" + expB + " (" + expbinB + ")
    Olvasott: " + readbinB);
    exec.Stop();
    return;
}

exec.Code("G00 G53 Y" + YToolChangePos); while (exec.IsMoving()) {}
exec.Code("G00 G53 Z" + ToolDown); while (exec.IsMoving()) {}
exec.Code("G01 F30 G53 Y" + YToolDockPos); while (exec.IsMoving()) {}
exec.Code("G04 P0.5"); while (exec.IsMoving()) {}

exec.Setoutpin(ToolUnclampPort, ToolUnclampPin); // UNCLAMP (nyit)
exec.Code("G04 P0.3"); while (exec.IsMoving()) {}

exec.Code("G01 F10 G53 Z" + (ToolDown - 0.05)); while (exec.IsMoving()) {}
exec.ClROUTpin(ToolUnclampPort, ToolUnclampPin); // CLAMP (fog)
exec.Code("G04 P0.3"); while (exec.IsMoving()) {}

exec.Code("G00 G53 Z" + SafeZ); while (exec.IsMoving()) {}
exec.Code("G00 G53 Y" + YToolChangePos); while (exec.IsMoving()) {}

// =====
// 3) VISSZATÉRÉS, OFFSZET
// =====
exec.Code("G00 G53 X" + Xoriginalpos + " Y" + Yoriginalpos); while (exec.IsMoving()) {}
exec.Code("G49"); exec.Wait(100); while (exec.IsMoving()) {}

if (!exec.Ismacrostopped())
{
    exec.Setcurrenttool(Newtool);
    exec.Code("G43 H" + Newtool);
    exec.Wait(100);
    while (exec.IsMoving()) {}
}
else
{

```

```
    exec.StopWithDeccel();  
    MessageBox.Show("Tool change was interrupted by user!");  
}
```