

```

// =====
// UCCNC M6 macro – 12 állásos esernyőtár (B tengely)
// + 4 bites abszolút tárkód olvasás/validálás
// + T0 → szerszám kirakás (csak drop, nincs felvétel)
// =====

// -----
// GÉP / I/O PARAMÉTEREK (KITÖLTÖTTEM A TE ADATAIDDAL)
// -----
const int MaxToolNum = 12;          // 12 fészek
const double ToolBOffsetDeg = 0.0; // ha T1 nem pont B=0°, itt adható eltolás

// UNCLAMP (tokmány nyit) – kimenet:
int ToolUnclampPort = 2;
int ToolUnclampPin = 14;

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

// 4 bites abszolút tárjel – BEMENETEK (a TE kiosztásod):
// bit1(LSB) = port 1 pin 2
// bit2      = port 1 pin 3
// bit3      = port 1 pin 4
// bit4(MSB) = port 1 pin 5
const int BitPort = 1;
const int Bit1Pin = 2; // LSB
const int Bit2Pin = 3;
const int Bit3Pin = 4;
const int Bit4Pin = 5; // MSB

// -----
// GÉPI KOORDINÁTÁK (ELLENŐRIZD!)
// -----
double SafeZ          = -0.5;
double ToolDown       = -5.339;
double ToolUp         = 0.0;
double YToolChangePos = -7.8;
double YToolDockPos   = -11.6913;

// -----
// B FÉSZEK-SZÖGEK (30°/fészek)
// -----
double[] ToolB = new double[MaxToolNum + 1];
double bstep = 360.0 / MaxToolNum; // 30°
ToolB[0] = 0.0; // park, ha kell
for (int i = 1; i <= MaxToolNum; i++)
    ToolB[i] = ToolBOffsetDeg + bstep * (i - 1);

// -----
// HELPER: bemenet olvasó
// MEGJEGYZÉS: ha nálad a függvény neve más (pl. GetInPinState),
// a lenti "exec.Getinpinsate" hívás nevét cseréld ki!
// -----
bool In(int port, int pin)
{
    return exec.Getinpinsate(port, pin); // ha más a neved: exec.GetInPinState(...)
}

// -----
// HELPER: 4 bites kód beolvasása (0..15)
// bit1 = LSB
// -----
int ReadCarouselCode()
{
    int b1 = In(BitPort, Bit1Pin) ? 1 : 0; // LSB
    int b2 = In(BitPort, Bit2Pin) ? 1 : 0;
    int b3 = In(BitPort, Bit3Pin) ? 1 : 0;

```

```

int b4 = In(BitPort, Bit4Pin) ? 1 : 0; // MSB
int code = (b1) | (b2 << 1) | (b3 << 2) | (b4 << 3);
return code; // 0..15
}

// -----
// HELPER: Várt kód adott fészekhez (a te táblád szerint)
// tool1 0001 (1)
// tool2 0010 (2)
// tool3 0011 (3)
// tool4 0100 (4)
// tool5 0101 (5)
// tool6 0110 (6)
// tool7 0111 (7)
// tool8 1000 (8)
// tool9 1001 (9)
// tool10 1010 (10)
// tool11 1011 (11)
// tool12 1100 (12)
// -----
int ExpectedCodeForTool(int tool)
{
    if (tool < 1 || tool > 12) return -1;
    return tool; // a fenti kiosztás pont 1..12 decimális
}

// -----
// HELPER: kód → fészek (reverse lookup), 1..12 vagy 0 ha érvénytelen
// -----
int CodeToTool(int code)
{
    if (code >= 1 && code <= 12) return code;
    return 0; // 0, ha nem mappolható
}

// -----
// HELPER: abszolút kód validáció adott fészeknél (B megállás UTÁN)
// Korrekció NINCS, csak hibaüzenet/Stop, ha nem egyezik.
// -----
bool ValidateCarouselAt(int expectedTool)
{
    exec.Wait(50);
    int code = ReadCarouselCode();
    int exp = ExpectedCodeForTool(expectedTool);
    if (code == exp) return true;

    // Formázott bináris kijelzés (4 bit)
    string Bin(int v){ return Convert.ToString(v, 2).PadLeft(4, '0'); }

    MessageBox.Show(
        "Carousel code mismatch!\n" +
        "Expected TOOL pocket: T" + expectedTool + " (code " + Bin(exp) + ")\n" +
        "Read code: " + Bin(code) + "\n" +
        "Állítsd be / ellenőrizd a jeladót, majd próbáld újra."
    );
    exec.Stop();
    return false;
}

// =====
// M6 MAKRÓ KEZDÉS
// =====
int Newtool = exec.Getnewtool(); // Tn
int Currenttool = exec.Getcurrenttool(); // aktuális (tokmányban lévő)

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

```

```

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

// Induláskori tár-állás beolvasása (információ/jövőbeli felhasználás)
int CarouselNow = CodeToTool(ReadCarouselCode()); // 1..12 vagy 0
// MessageBox.Show("Carousel jelenleg T" + CarouselNow);

// Currenttool normalizálás
if (Currenttool < 0 || Currenttool > MaxToolNum) Currenttool = 0;

// *** T0 → csak kirakás kérése ***
bool onlyDrop = (Newtool == 0);

// Kizárólag 1..12 közötti felvételt engedünk (ha nem T0)
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 álljon, Z fel
// -----
double Xoriginalpos = exec.GetXmachpos();
double Yoriginalpos = exec.GetYmachpos();

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

exec.Setoutpin(DustCollectorPort, DustCollectorPin); // ha használod

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

// -----
// 1) LEDOBÁS (ha van a tokmányban valami)
// -----
if (Currenttool != 0)
{
    exec.Code("G00 G53 B" + ToolB[Currenttool]);
    while (exec.IsMoving()) {}
    if (!ValidateCarouselAt(Currenttool)) 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
    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 → itt megállunk (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.StopWithDecel();
        MessageBox.Show("Tool change was interrupted by user!");
    }
    return;
}

// -----
// 2) FELVÉTEL (ÚJ SZERSZÁM)
// -----
exec.Code("G00 G53 B" + ToolB[Newtool]);
while (exec.IsMoving()) {}
if (!ValidateCarouselAt(Newtool)) 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.Setoutputpin(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.StopWithDecel();
    MessageBox.Show("Tool change was interrupted by user!");
}

```