

VB .Net AutoLeveller

Ha valaki használ Eagle -> pcb-gcode -> Autoleveller -> UCCNC kvartettet, akkor tapasztalhatja, hogy nem zökkenőmentes az együttműködésük. Először is, a pcb-gcode (<http://pcb-gcode.org/list.php?12>) nem pont olyan kódot generál, ami tetszene az UCCNC-nek. Ezt aránylag egyszerű megoldani: létre kell hozni egy profilt az UCCNC-nek. Ehhez a Mach3 profilt érdemes lemásolni, majd a megfelelő helyeken módosítani kell. Mielőtt bárki elkezdene gépelni, minden file megtalálható a zip-ben.

1. A várakozás paramétert megjegyzésbe kell tenni, mivel az UCCNC úgyis kezeli a motor ki- és bekapcsoláskor a várakozásokat:

```
string DWELL = "(-G04 " + PARAM + "%f)" + EOL;
```

2. A G82-t sajnos nem értelmezi az UCCNC, így G83-at kell megadni, de ennek más a paraméterezése (ezt kompletten kimásolom):

```
//  
// Drilling holes  
//  
// G83 Xx.xxx Yy.yyy Zz.zzz Fff.f Qq.qqq Rr.rrr, no #dwell parameter!  
//  
string DRILL_CODE      = "G83 ";  
string RELEASE_PLANE   = "R" + FORMAT;  
string DWELL_TIME      = PARAM + "%f";  
string DRILL_PAR_Q     = "Q5.0 ";  
string DRILL_FIRST_HOLE = DRILL_CODE + MOVE_XYZ + DRILL_PAR_Q + FR_FORMAT + RELEASE_PLANE +  
    COMMENT_BEGIN + "-" + DWELL_TIME + COMMENT_END + EOL;  
string DRILL_HOLE      = DRILL_CODE + MOVE_XY + DRILL_PAR_Q + EOL;
```

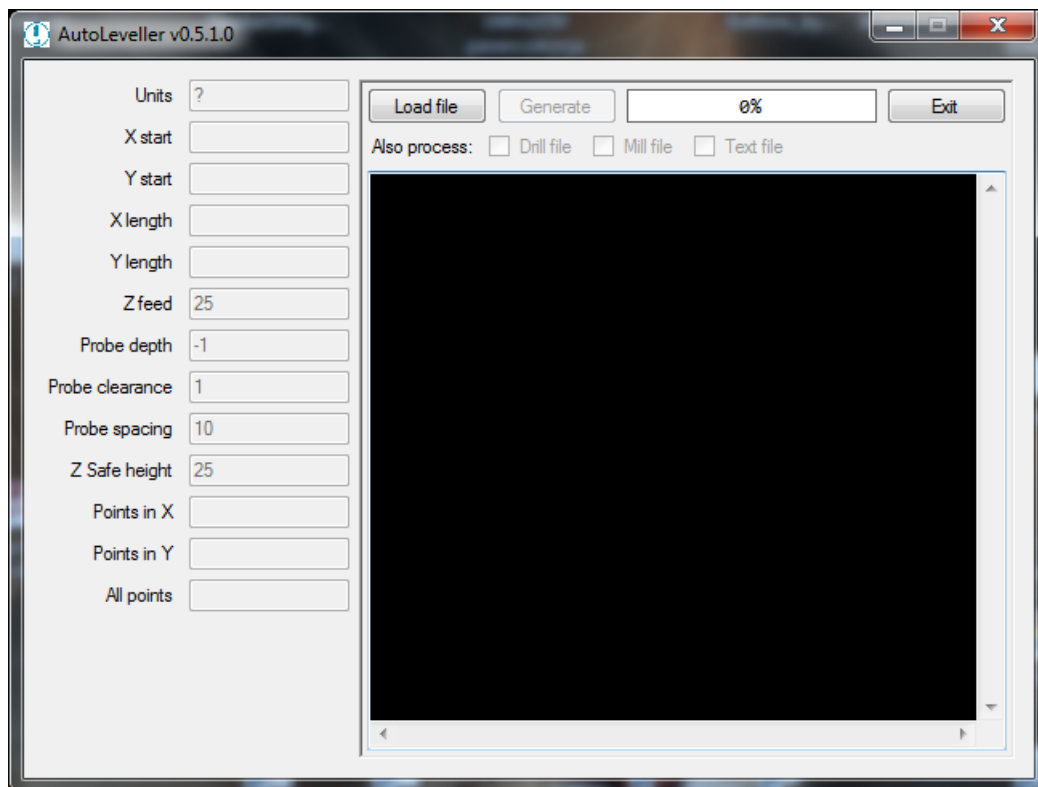
3. A szerszámcserébe is bele kell nyúlni a megjegyzés formátuma miatt:

```
string TOOL_CHANGE     = OPERATOR_PAUSE + TOOL_CODE + COMMENT_BEGIN + "Diameter: " + FORMAT +  
    COMMENT_END + EOL;
```

Az ezzel a módosított profillal generált kódot már megveszi az UCCNC. (A megjegyzések "(-" kezdetéről majd később.)

Létezik egy okos találmány, az Autoleveller (<http://www.autoleveller.co.uk>). Panel készítéséhez szerintem elengedhetetlen. Dióhéjban annyit csinál, hogy leméri a panel felületét, majd az így kapott adatokkal módosítja a marást a Z tengely mentén.

Az egyik probléma az, hogy ő sem ismeri az UCCNC lehetőségeit. A másik, hogy a pcb-gcode külön generálja a marás (etch), a fúrás (drill), a kivágás (mill, 46-os réteg az Eagle-ben) és a szöveg (text) rétegek g-kódját, az Autoleveller pedig úgy írja át a kódot, hogy először mér, majd ennek felhasználásával mar. Ez az első menetben rendben is volna, de miután kimarta a vezetősávokat, már nem tud mérni a szokásos érintkezős módszerrel. A harmadik probléma pedig az, hogy ugyan letölthető az Autoleveller egy régebbi változatának a forrása, de java-ban íródott. Dicséretes tény, hogy ennek köszönhetően linuxtól MacOS-ig mindenben elfut, de bő 30 év programozás alatt sem tudtam a java-val megbarátkozni, ezért úgy döntöttem, hogy az egészet újraírom a saját elképzeléseim szerint. Így született meg a VB .Net AutoLeveller.



Alapképernyő

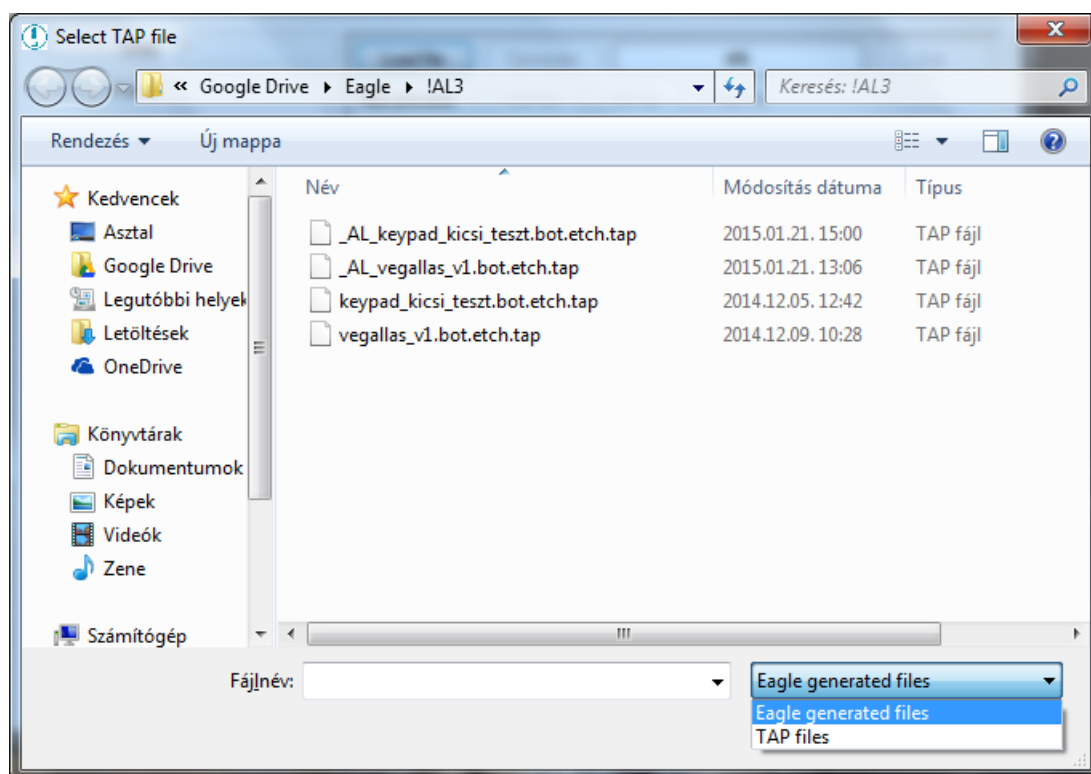
Az egyik legfontosabb eltérés az eredeti programhoz képest, hogy az én változatom szétválasztja a mintavételezést és a tényleges munkavégzést. A mintavételezés végén a kapott Z eltéréseket automatikusan elmenti, így egyrészt később is lehet folytatni a munkát, másrészt, ha esetleg valami probléma miatt újra el kell indítani a marást, akkor nem fog újra mintavételezni, hanem simán rá lehet engedni újra a munkadarabra. Amiért még nagyon fontos ez: a fúráshoz és maráshoz, esetleges feliratozáshoz is ugyanezt a mintavételezési adathalmazt használjuk, így egy méréssel mindent meg tudunk oldani.

Ahhoz, hogy a mintavételezés eredményét tudjuk tárolni és visszatölteni, el kellett készíteni néhány makrót az UCCNC-hez. Ezek a következők:

- M310.txt: a G31-gyel mért Z pozíciót beteszi a #9 változóba, innen már kódból is tudjuk kezelni;
- M311.txt: #8-ban be kell állítani az utolsó mentendő változó sorszámát, a makró #500-tól ezzel bezárólag menti a változókat egy file-ba. A mintavételező kód generálásakor a #8 értékét a g-kódba beírja a generátor. Az elkészült file kiterjesztése ".m311" lesz, az eleje pedig a futó g-kód neve.
- M312.txt: a betöltött g-kód neve alapján kitalálja az ".m311" file nevét, majd ezt a file-t felolvassa és az abban tárolt változókat beállítja. Ha az adott könyvtárban létezik "common.m311" nevű file, akkor azt tölti be.

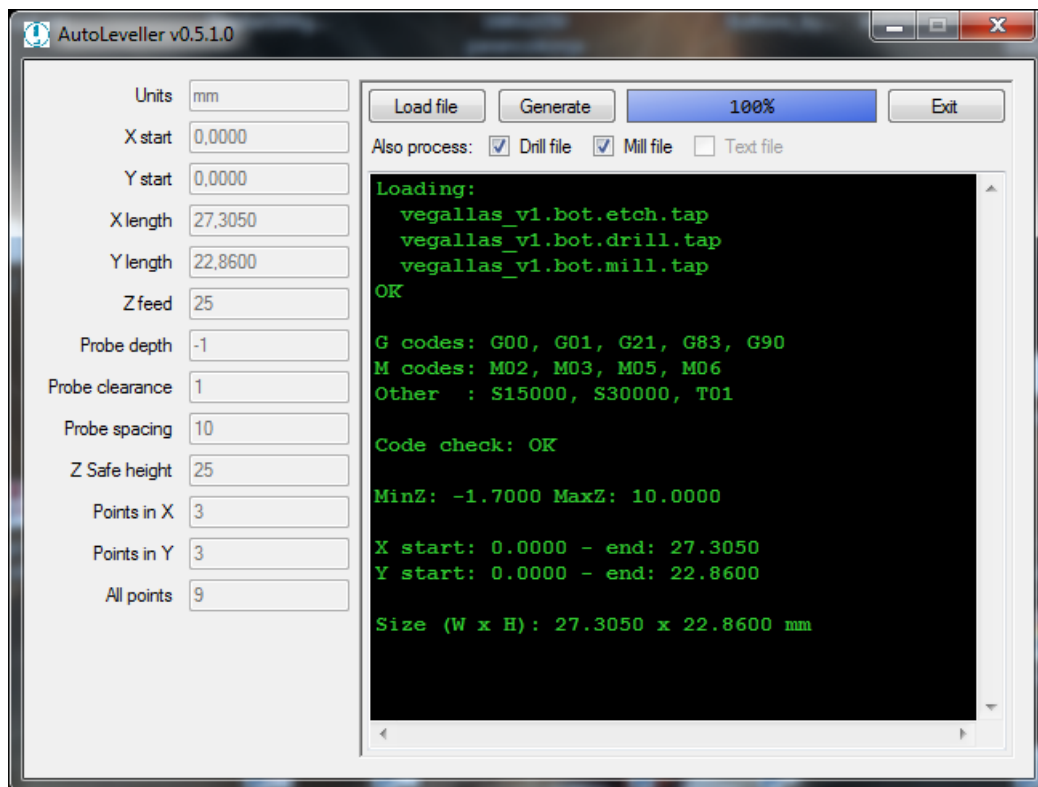
A program kifejezetten a pcb-gcode által generált kód feldolgozására készült, ennek ellenére jó eséllyel meg lehet vele etetni más kódot is. A bemenő kódot a pcb-gcode által generált file nevekre terveztem, ahol a nevek így képződnek: a rajz eredeti neve után ".bot" vagy ".top" a rétegtől függően, majd ".etch", ".drill", ".mill" és ".text" attól függően, hogy mit csinálunk, végül a kiterjesztés, ami ".tap". Ez azért fontos, mert alapértelmezés szerint azokat a file-okat látjuk a Load gombra kattintva, amelyek ".etch.tap"-ra végződnek. (Át lehet állítani "*.tap"-ra.) Ha a file-t kiválasztottuk, akkor megnézi, hogy a többi végződés létezik-e. Ha

igen, akkor a megfelelő checkbox-ok engedélyezetté válnak, alapértelmezésben bepipálva. Eldönthetjük, hogy a szintezést mely file-okon végezze el a program.

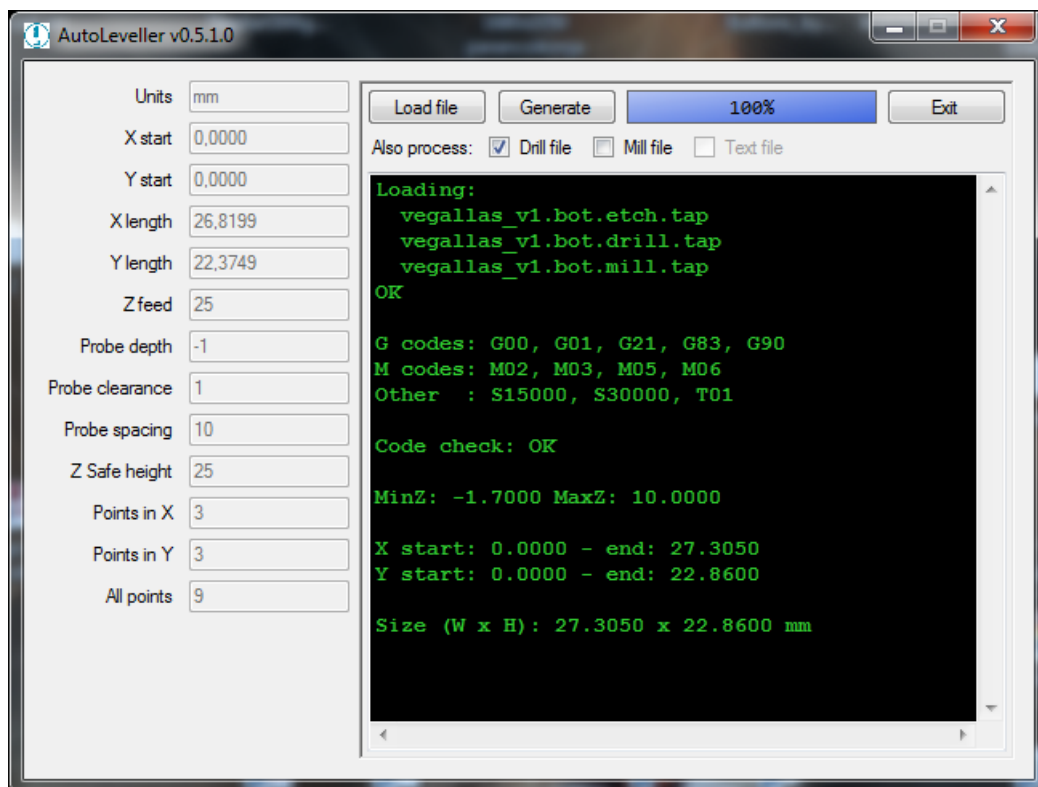


File megnyitás, alapból az .etch.tap végűek jelennek meg

A file kiválasztásakor a program felolvassa a g-kódokat, értelmezi azokat, és kiszámolja, hogy mekkora a munkaterület. A munkaterületet úgy határozza meg, hogy az összes kiválasztott file által lefedett területen mintavételezzen. Ha tehát például az ".etch"-ben 1.1,0.8-tól 42.45,23.776 a feldolgozott terület, de a ".mill"-ben 0.0,0.0-tól 45.0,25.0, akkor a nagyobbik lesz érvényben. Ha kivesszük a pipát a ".mill" elől, akkor a bal oldali mezőkben láthatjuk, hogy a terület is kisebb lesz, ezt a területet fogja felosztani mintavételezéshez.

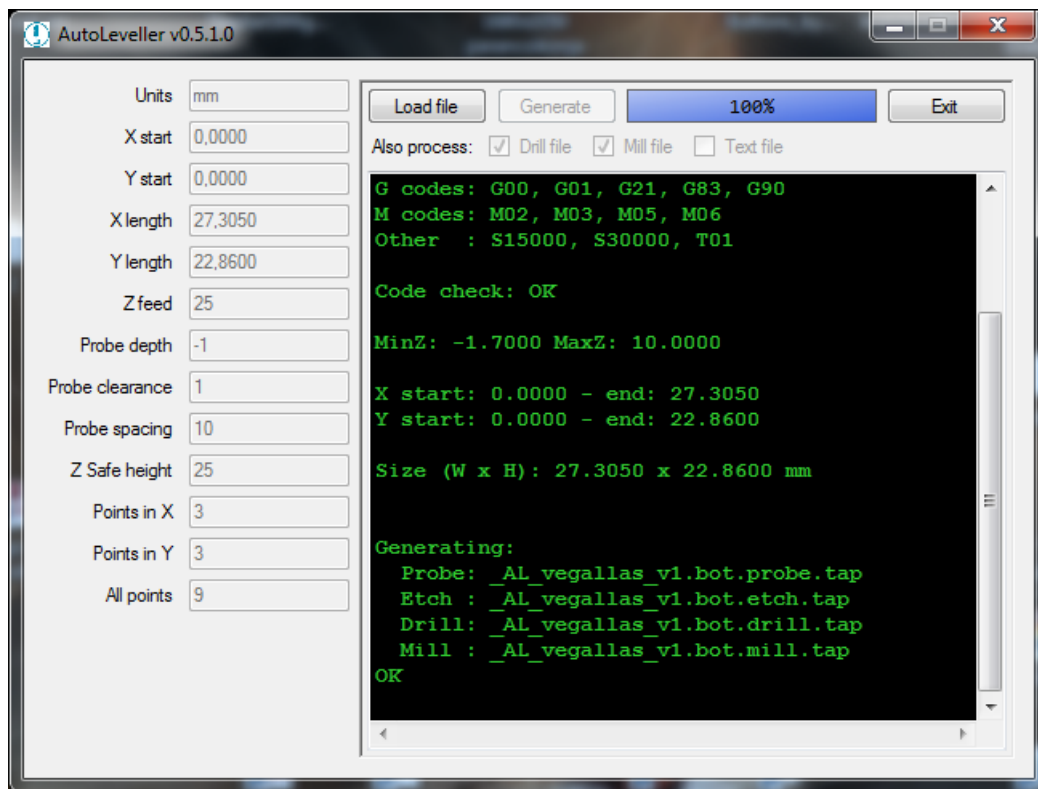


File betöltve, minden kijelölve



File betöltve, de mill nélkül

Generálás indításakor először létrehozza a mintavételező g-kódot ".probe.tap" végű névvel, majd a többi bejelöltet is. Mindegyik file az eredeti könyvtárban jön létre, a nevük eleje "_AL_" lesz.

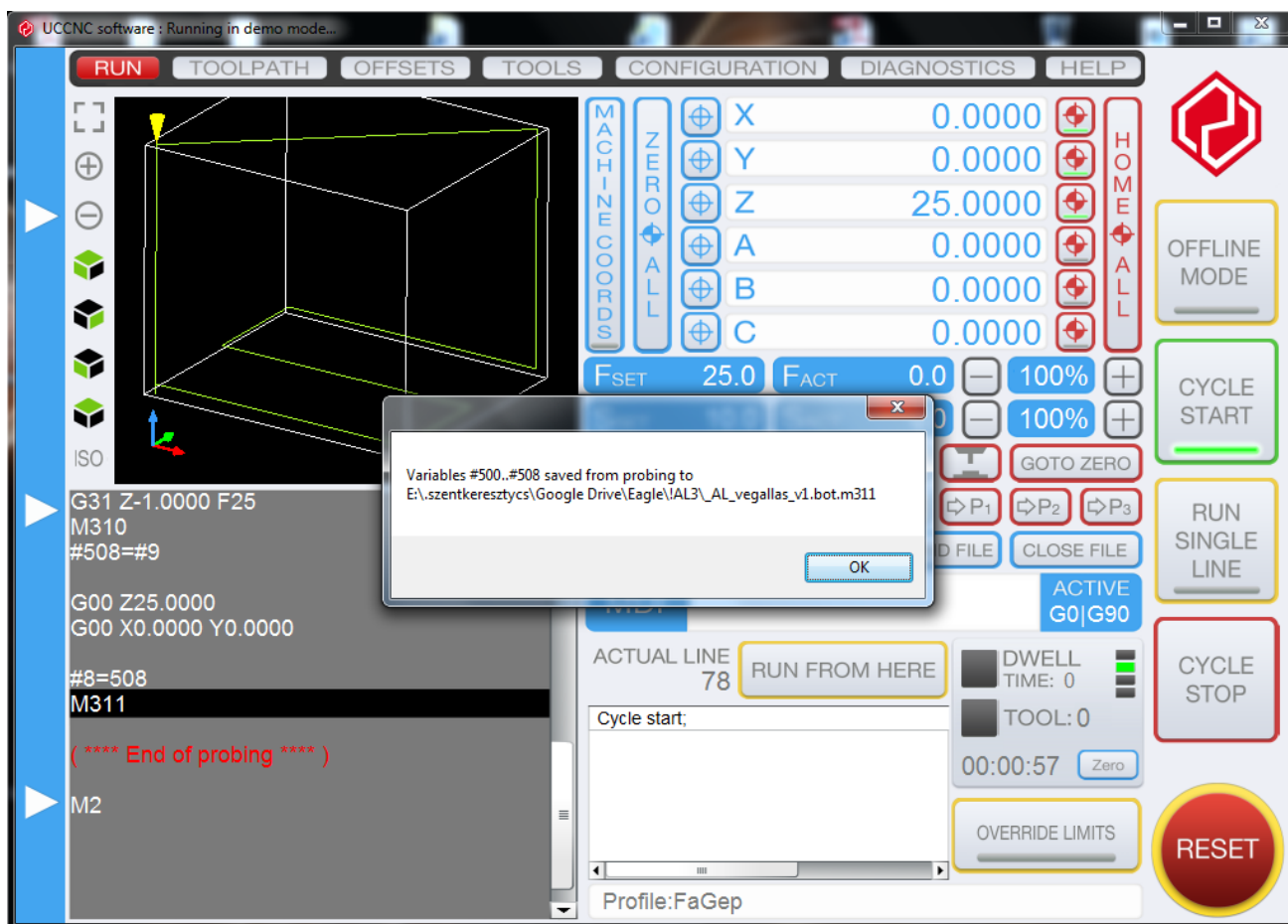


Generálás után

Már csak be kell tölteni az UCCNC-be, mégpedig először a ".probe.tap" végűt. A mérés végén kapunk egy jelzést, hogy az adatok hová kerültek.

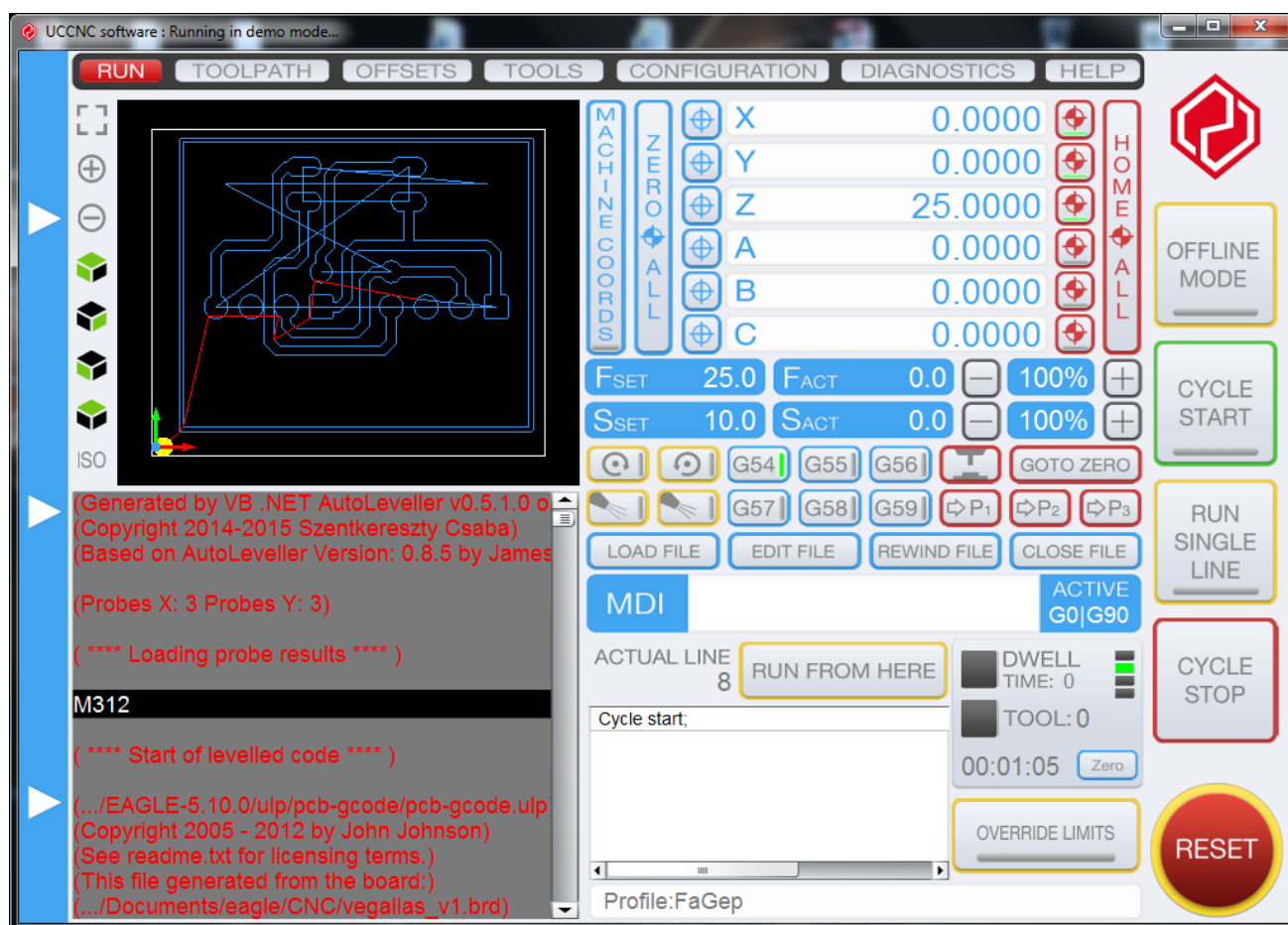
Korábban már említettem a "(-" kezdetű megjegyzéseket. A generátor az így jelölt megjegyzéseket nem emeli át az eredeti kódból, így kicsit olvashatóbb lesz az eredmény. Azért van erre szükség, mert a pcb-gcode-ot úgy akartam módosítani, hogy csak a profil file-ban legyen változás, így újabb verzió megjelenésekor nem kell másra figyelni. A pcb-gcode által kiszámolt értékeket valahová kötelezően be kellett írni: irány egy megjegyzés, amit később ki is veszek a kódból.

Még egy-két megjegyzés. Az eredeti file-okhoz soha nem nyúl hozzá a generátor, így bármikor újra lehet futtatni egy esetleges kézi módosítás után. A program elég érzékeny a file-nevekre, ezért próbáljuk az eredeti koncepciót követni. Ha máshonnan származó g-kódot szeretnénk szintezni, akkor érdemes úgy átnevezni, hogy a file neve ".etch.tap"-ra végződjön. Kezdetben volt olyan hiba, hogy a bemenő file-t lerombolta, azóta szigorúbb a nevek kezelése, cserébe előfordul, hogy file-név hibát ír a státusz ablakba és nem futtatható. Átnevezés a megoldás.



Lefutott a mintavételezés

Következő lépésben mehet a többi file is.



Indulhat a marás (M312 fogja betölteni az adatokat)

FIGYELEM! Ez a program még nem érte el az 1-es főverziót. Hibák lehetnek (és gondolom, vannak is) benne, ezért ellenőrizzük, hogy mit csinál a gép. Ez a program ingyenes. Az esetleg elrontott munkadarabokért, netán egyéb káresetért nem vállalok felelősséget.

2015. 01. 22. Szentkereszty Csaba