**Snake(kígyós játék)**

**Lukács Botond**

**Számítástechnika IV. A**

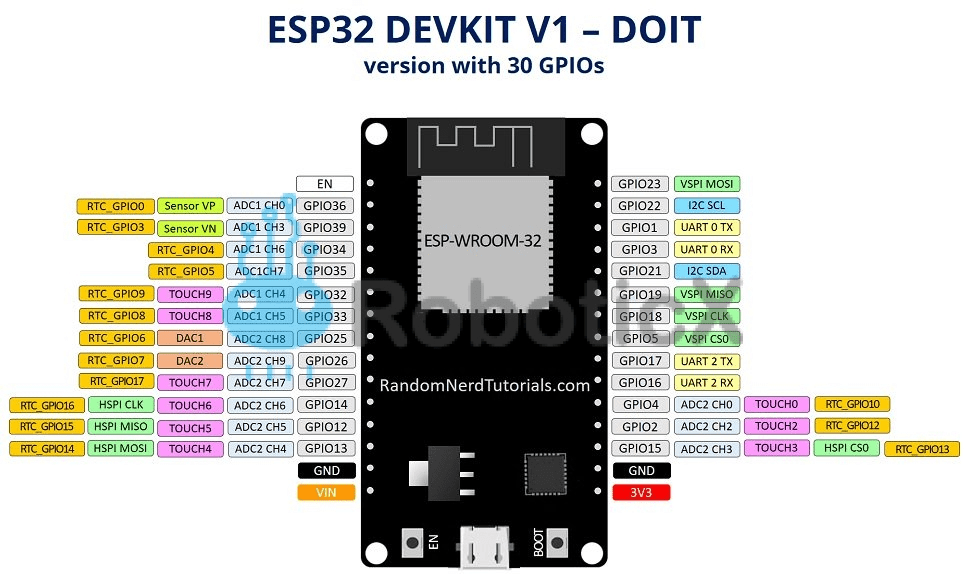
**Bevezető**

**A projekt megvalósítására egy régi telefonos játékotgondoltam ki, amelyet 4 iránygombbal lehet vezérelni és egy kijlzőn látható a játék jelenlegi állapota. A játék lényege, hogy a kígyót a jutalmakhoz vezessük, amitől a kígyó hosszabb lesz és a cél, hogy minél nagyobb kígyót létrehozni annáélül, hogy saját magunkba vagy a képernyő szélébe ütköznénk. A játék színeit leht változtatni a programkódbeli lonstansok módosításával. A játék pálya mérete egy 30\*40-es pálya és a játékos egy 2 hosszú kígyóval indul.**

**Felhasznált eszközök**

**ESP**

**A játék vezérlését egy ESP-WROOM-32-es vezérli ami egy fejlesztő lapra van szerelve ami egy ESP32-DEVKIT V1 - DOIT verzió, ennek 30 GPIO lába érhető el külsőleg. A ESP chippet egy ilyen lapra szerelésének a lényege, hogy a lábak sokkal egyszerűbben elérhetőek, mival most már breadboard-on is használható a rendszer ami künnyű prototizálásra ad lehetőséget és nem kell minden változtatásért új PCB-t készíteni. A másik fontos tulajodonsága, hogy lehetővé teszi, hogy egyszerűen USB-n keresztül programozzuk és tápláljuk az ESP-t így nem kell a feszültség átalakítással és a programozással bajlódni. Viszont sok ilyen lapnak máshol nehetnek a végső kivezetései, így csatolom lábkiosztását az én verziómról.**

****

**Kijelző:**

**A kijelzőnek egy ILI9341-et választottam, legfőképpen mivel már ezelőtt rendelkeztem ezzel az eszközzel. Ez egy 2.4” képernyő átlójú és 320x240-es felbontású kijelző ami 16 bit-es színes képet képes megjeleníteni. A kommunikáció SPI-n keresztül történik, maximális SPI órajele nagyobb, mint amit az ESP létre tud hozni így nem lehetséges olyan probléma, hogy az órajel túl magas és a képernyő nem képes stabilan olvasni az adatot az SPI-n keresztül. Belső memóriával rendelkezik így csak akkor kell adatot küldeni, amikor meg kell változtatni a képet, egy konstans kép esetén elég egyszer elküldeni az adatot és a kép fog létszani a képernyőn még az Spi vezetékek eltávolítása után is.**

**A kép kijelzésére több módszer is lehetséges, a 2 legnépszerűbb a partial area és a full area. Partial area esetén meg kell előre adni, hogy a képernyő mely részén dolgozunk, majd folytonosan küldeni az adatot és a képernyő belsőleg elintézi, hogy a számlálót növelje a következő kéőkocka memória címére amikor egy adat megérkezett. Ez leginkább kis teljesítményű mikrovezérlőkön, pl Arduino-k használják, mivel ott nem fér el egy teljes memória buffer a memóriában, hogy egyszerre kirajzolja a teljes képet. A másik a full area, itt folytonosan küldjük az adatot és a teljes képernyőn pixelről pixelre megadjuk, hogy mi milyen színű. Az adat bemaneti sebesség nem számít, mivel ha egy képkocka alatt nem adjuk meg az összes képkocka adatját akkor is kirajzolja amellyire megvan.**