

# Gépészmérnöki alapismeretek projekt munka Autós óracsoportból szimulátorkijelző

Bacsi Levente

2020.11.22.

Gépészmérnöki alapismeretek projekt munkámban már a címben is említett eszközt készítettem el, ezt szeretném részletesebben is bemutatni. Azért esett a választásom erre, mert elég jól fedi érdeklődési területemet, és az elkészült munkát is tudom hasznosítani a későbbiekben. Továbbá a megvalósítás szinte semmibe nem került, ezt majd részletesebben is taglalom.

De mi is pontosan ez a projekt? A címben már tulajdonképpen összefoglaltam, de ennyire azért nem egyszerű. Először is meg kell találni a megfelelő márkát és típust, ami elektronikus műszerezettséggel rendelkezik (tehát a sebességmérő nem lehet spirálos), azonban nem használ még CAN-busz szabványt. Ezután kell egy mikrovezérlő, amivel kapcsolatot lehet létesíteni az órák, és a számítógép között. Szükség van a műszer tápellátására is, és mivel az autó elektromos rendszere általában 12 voltos, ezért egy ennek megfelelő tápegységet kell választani. Nem szabad megfeledkezni a mikrovezérlőt és az óracsoportot összekötő jumper kábelekről, illetve a szoftveres csomagot is el kell készíteni.

## 1. Az óracsoport

Mivel ez lesz a projekt alapja, ezért fontos, hogy megtaláljuk a megfelelőt. A feljebb taglalt szempontok mellett figyelembevettem még sok minden mást is. Fontos volt, hogy a fordulatszám mérő viszonylag nagy tartományt lefedjen, ezért dízel járművéké kiesett. Emellett számított még a sebességmérő is, hogy aránylag magas legyen megjeleníthető érték. Ismerni kell továbbá a műszerek lábait, hogy mikrovezérlővel összekötve a megfelelő pineket, fel tudjuk azt programozni.

Az interneten keresgélve a BMW E36 típusának (3-as BMW 1990-2000-ig) órájára esett a választásom, mivel minden kritériumnak megfelel, és már is használta más is hasonlóra, így vannak tapasztalatok ezzel kapcsolatban. Azonban ezek között sem mindegy melyik, hiszen létezett dízel, négy- és hathengereshez való is. Kis kutatás után kiderült, hogy nekem egy hathengeres benzines variánsra van szükségem, mivel a fordulatszám mérője 7000 RPM-ig, a sebességmérő pedig 260 km/h-ig van skálázva.



1. ábra. A beszerzett óracsoport

Szerencsére több ismerősömmnek is van ilyen járműve, egyikük pedig teljesen ingyen megszánt egy ilyen eszközzel. Mivel a km-számláló folyadékkristályos kijelzője meg van folyva, ezért közúti forgalomban lévő járműbe már nem alkalmas (műszaki vizsgán bukótétel, ha leolvashatatlan a km-óra állás), de az én felhasználásomra megfelelő lesz.

## 2. A mikrovezérlő

A projekt "agya". Feladata, hogy a játékszoftverből kinyert adatokat a műszerek számára értelmezhetővé tegye, ezt pedig a rajta futó program segítségével fogjuk elérni.

Nagyon-nagyon sokféle mikrokontroller létezik jelenleg a piacon, ezek közül kell kiválasztani a számunkra optimálisat. Számítási teljesítményben, csatlakozókban, és természetesen árban is nagy a szórás, és ebből a széles választékból kell megtalálni a mi szempontjainknak megfelelőt. Nekem jelenleg elsődleges volt az Arduino-kompatibilitás, és a kis méret. Nagy számítási teljesítményre nincs szükség, ahogy kimenetből is elég 2 PWM, és 2 "sima" digitális. Ezeket tökéletesen teljesíti egy alap Arduino Uno, vagy egy Nano. A méret miatt azonban utóbbira esett a választásom, illetve van is nekem ilyenem, kínai kivitelben. Ez tulajdonképpen csak árban tér el az eredeti Arduinótól (2-3 dollárért beszerezhető), a processzor és a lábkiosztás megegyezik, illetve a fejlesztői környezettel is kompatibilis. Lehet egyébként kapni lábak nélkül is, többféle chippel, de ez a variáns (Atmega 168P alapú, forrasztott lábakkal) tökéletesen megfelel a feladatra.



2. ábra. Atmega 168P alapú mikrovezérlő

Szükség van még egy micro-USB kábelre a számítógéppel való összekötéshez, valamint a pinekre illeszkedő kábelekre, amikkel az órára lehet csatlakoztatni a vezérlőt. A PC-re le kell tölteni az Arduino IDE-t (ez egy integrált fejlesztői környezet), ami tartalmazza a szükséges illesztőprogramokat is, valamint ezzel tudjuk módosítani és feltölteni a programot a boardra.

## 3. Tápellátás

A következő megoldandó feladat a tápellátás megtervezése.

Mint nagyjából az összes elektromos berendezés a gépkocsikban, a műszeregység is 12 voltos hálózatról működik. Figyelembe kell venni azt is, hogy több szervomotort, izzókat kell ellátni árammal, így nem érdemes túl gyenge tápegységet választani. Én egy csatlakozót is készíték, hogy ne kelljen az egész eszközt táppal, meg a hosszú vezetékkel mozgatni.

Személy szerint én nagyon szeretek régi, elfelejtetett, másra nem jó dolgokat felhasználni (kicsit már maga a projekt is ilyen), a tápellátás kérdéskörében is valami hasonló megoldást kerestem. Sikerral is jártam, az egyik fiók mélyén találtam egy régóta nem működő LED-szalagot. Ennek 3 része van, a táp, egy kis vezérlődoboz, illetve maga a szalag, és persze 12 volttal működik. Két alkatrészt használtam fel belőle, a tápegységet (2 Ampert tud, remélhetőleg elég lesz), illetve leforrasztottam a vezérlőjéről a csatlakozót, erről később részletesebben.



3. ábra. A felhasznált 12 voltos tápegység

## 4. Szoftver

A hardvereket már áttekintettük, de legalább ennyire fontos része a projektnek a szoftver.

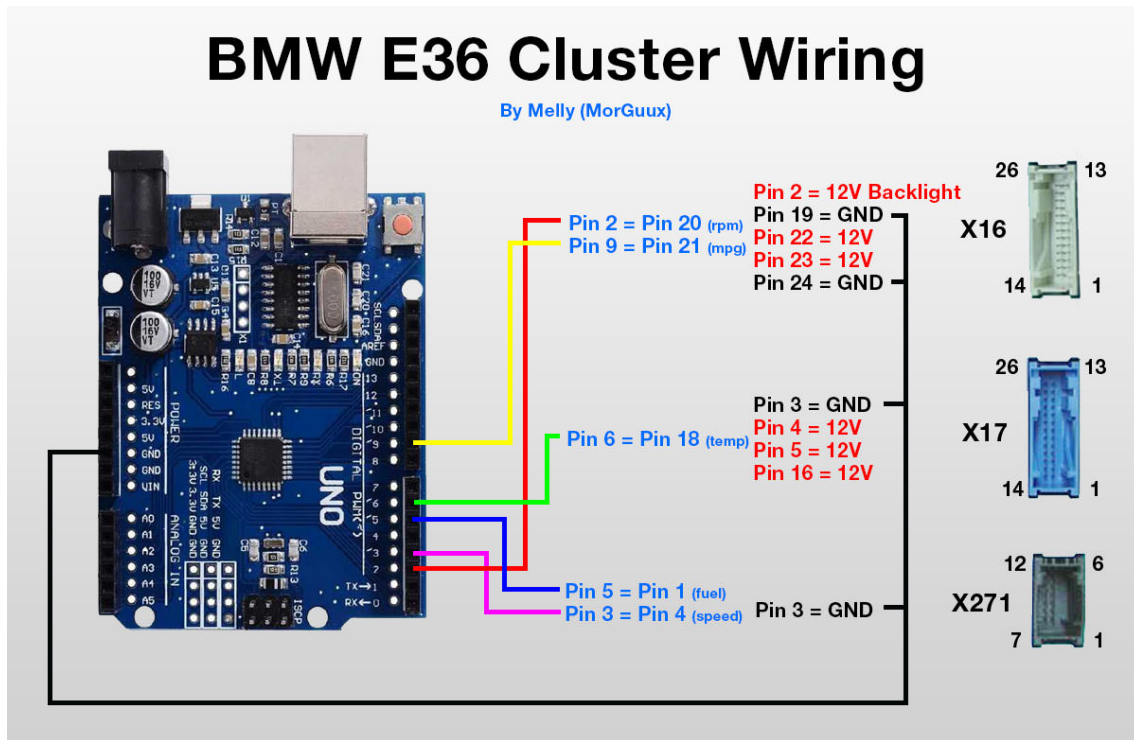
Szerencsére viszonylag jól kitalált és megtervezett termékek léteznek erre, mivel (főleg a vírus óta) elég nagyot nőtt a szimulátorozás, mint hobbi népszerűsége. Leginkább olyan szoftverek ezek, amiket nem nagy cégek, hanem egyszerű emberek fejlesztenek. Ennek megfelelően nem mindig tökéletesek, de teljes mértékben használhatók.

Az egyik ilyen a Simhub, most is ezt fogom használni. Az elmúlt években elég komoly fejlődésen ment keresztül, egyre több minden került bele. Képes kezelni külső kijelzőket, okostelefonokat, Arduinokat. Ezek közül én a legutóbb fogom használni, valamint a fejlesztői környezetet. Az IDE nagyon fontos szoftver, hiszen ez fordítja a magasabb szintű programozási nyelvről a gép számára is értelmezhetőre a kódot, illetve Arduinónál a PC és a mikrovezérlő közti kapcsolatot is megteremti.

Játékból nagyjából bármit választhatnék, mivel minden népszerűbb szoftvert támogat a program. Én általában az rFactor, és Assetto Corsa játékokat használom, most is ezek kerülnek sorra. Hogy működjön minden, ezeket a szoftvereket is módosítani kell minimális mértékben, de erről majd később.

## 5. Összeállítás

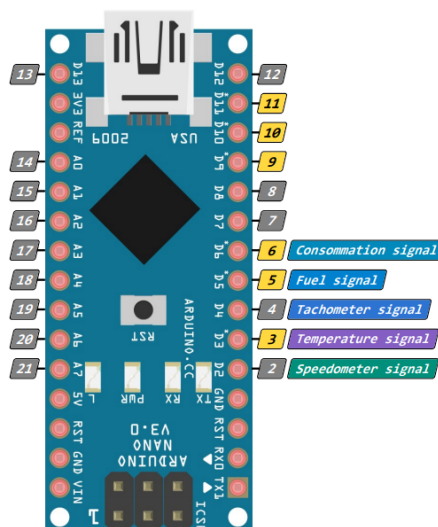
Az összeszerelés során először a megfelelő lábak megkeresésével foglalkoztam. Ezen az óracsoporton 3 csatlakozó van, 2 db 26, és 1 db 12 pinos. Minden műszernek megvan a 12 voltos pozitív és negatív lába, valamint az adatokért felelős vezetéke. Ezeket nem egyszerű megkeresni, ha ez embernek semmiféle dokumentációja nincsen, de szerencsére az interneten találtam bekötési rajzot.



4. ábra. Forrás: <https://github.com/SHWotever/SimHub/wiki/BMW-E36-Cluster-Setup>

Alapvetően ezen ábra alapján dolgoztam, azonban én a negatív vezetékeket a tápcsatlakozó negatív pólusára kötöttem. Először a megfelelő lábakra csatlakoztattam őket, majd elvágtam, és blankoltam a kábelek másik felét, ezeket pedig összesodortam. Nem szabad megfelekedezni az Arduino GND vezetékéről sem, azt is össze kell kötni a tápcsatlakozóval. Ehhez hasonlóan készülnek el a pozitív vezetékek is.

A többi vezetéknél már bonyolultabb, hiszen az Arduino lábkiosztását is ismerni kell. Ez is megtalálható az interneten, ez alapján jutottam arra, hogy a 3, 5, 6, 9, 10, 11 pinnek szolgáltatnak PWM kimenetet, ezért a vízhőfok, és az üzemanyagszintjelző is ezek valamelyikére kell kerüljön. A fordulat -és sebességmérő tulajdonképpen bármelyikre kerülhet, de azért célszerű nem-PWM pinre csatlakoztatni. Be lehet kötni a fogyasztásmérőt is, azonban az Arduino korlátozásai miatt ez csak a két végállás között tud mozogni. Ezek ismeretében a SimHub varázslójával már elkészíthető az Arduino-ra szánt kód, egy viszonylag egyszerű grafikus felületen.

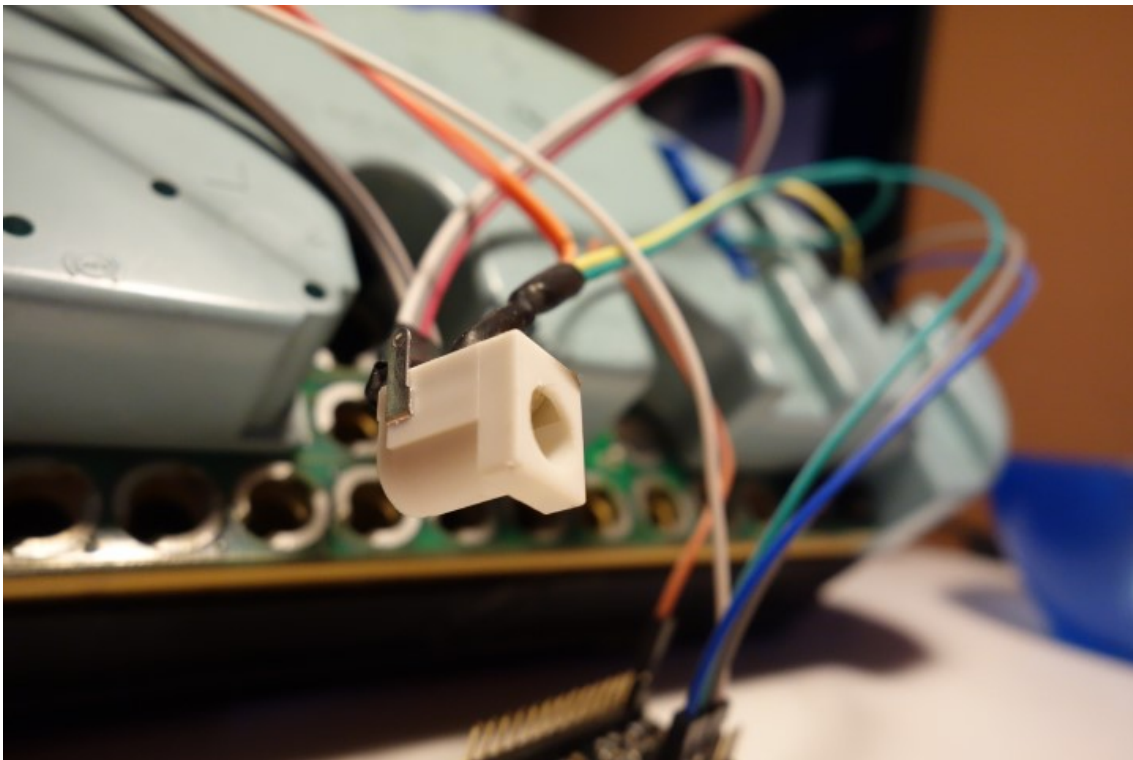


5. ábra. Az Arduino lábkiosztása

Szerencsés esetben már fel is lehetne tölteni a kódot a mikrovezérlőre, azonban az általam használt darabot nem támogatja teljesen a SimHub, mint kiderült. Szerencsére erre is van megoldás, a generált programot kell megnyitni az Arduino IDE-ben, és ott már át lehet állítani a processzor típusát sokkal több félére. Itt kell kiválasztani az Eszközök fül alól az Atmega 168-ast. Innentől már csak csatlakoztatni kell USB-n az eszközt, és az IDE lefordítja, majd feltölti a kódot.

Szoftveresen már készen vannak az alapok, itt az ideje annak, hogy a hardver is elkészüljön. Mint említettem, egy nem működő LED-sor lesz a donor a tápellátáshoz, első körben ezt kell szétszedni. Kipróbáltam egyébként több itthon talált tápegységet is, de úgy tűnik ez a 2 Amper mindenképpen szükséges hozzá, hogy minden megfelelően működjön, illetve azokat direktben kellett volna forrasztanom, a csatlakozós pedig sokkal jobban tetszett. Visszatérve a témához, a kis LED-sor nem egy bonyolult szerkezet, a műanyag házát kellett szétpattintanom, majd a nyákról leforrasztani a tápcsatlakozót. Ez egy egyszerű, hengeres kialakítású csatlakozó, a belső pin a pozitív, a henger "palástja" pedig a negatív.

Ezek ismeretében már csak rá kell forrasztani az összetekert vezetékeket a tápcsatlakozóra. Szerencsére ez egy elég könnyű feladat, még egy forrasztásban nem túl jártas embernek is (mint én). Érdemes szigetelni is a kábeleket, már csak az esztétikum jegyében is. Én zsugorcsovot használtam ehhez, hiszen mégiscsak egy igényesebb és tartósabb megoldás mint a szigetelőszalag, de tulajdonképpen az is megteszi.



6. ábra. Az elkészült tápcsatlakozó

## 6. Utolsó simítások, használat

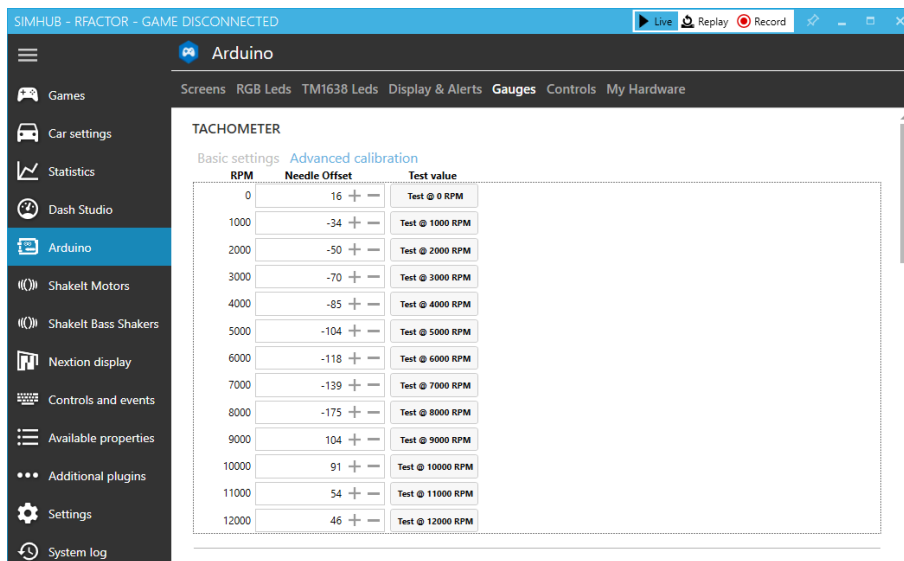
Miután nagyjából minden elkészült, már csak a finomhangolások vannak hátra.

Elsőként be kell másolni egy .dll fájlt a játék főkönyvtárába, ami az adatok eljuttatását fogja szolgálni a SimHubig. Ez sem egy bonyolult művelet, innentől viszont fáradalmasabb lépések jönnek.

Be kell állítani és kalibrálni a mérőórákat, amire elég sok opció, és beállítás van. Mindközül talán a sebességmérő a legegyszerűbb, itt egyedül csak a legkisebb és legnagyobb megjeleníthető értéket kell megadni, majd tesztelni. Nekem ez elsőre működött, 0-260-ig mident tökéletesen jelez, és a km-számlálót is pörgeti.

A fordulatszám-mérő kicsit bonyolultabb, itt nem csak az óra tulajdonságait kell megadni (mérés-határok), hanem van ezresenként is lehet (érdemes is) kalibrálni az egyes fordulatszám-értékeket. Ez viszonylag hosszadalmas, de csak

egyszer kell megcsinálni. Fontos opció még, hogy a műszer méréshatárához arányosan mutassa-e a fordulatszámot (ez akkor érdekes, ha mondjuk valamilyen rendkívül magas fordulátú motorral rendelkező autó van szimulálva), én ezt is bekapcsoltam. A fogyasztásmérő is működik, természetesen pontos értéket nem tud adni, de a fordulatszám növekedésére ugyanúgy kileng, mint valójában.



7. ábra. Végtelen állítási, és szöszölési lehetőségek

A vízűfok és az üzemyangszint már kicsit bonyolultabb kérdés, ezek alapvetően is más elven működnek, PWM-el (a feszültséget nagyon gyors ki-be kapcsolásokkal szabályozzák), valamint nem minden játék szimulálja ezeket. Az üzemyangszintet sikerült a SimHubban működésre bírnom, a játékban azonban nem mindig működik. Érdekes hiba, azonban nem egyedi az eset, még nem teljesen sikerült megfejteni az óra működését. A vízűfok az enyémen viszont valamiért abszolút nem működik, vagy egy nagyon magas, vagy egy nagyon alacsony értéken áll meg, és nem mozdul. Lehet hogy az óracsoportom hibás, vagy csak kijöhetett a kínai Arduino hátránya (lehet hogy rosszul kezel több PWM-kimenetet)? Egyelőre sajnos nem tudom, de továbbra is keresem a hiba okát, remélem sikerül majd rájőnnöm, és teljesen befejezmem a szerkezetet.



8. ábra. Eddigi élete során nem sokat állt így

## 7. Összegzés

Összegezve a projektet, én jól éreztem magam a készítése közben. Volt ugyan pár fárasztóbb és/vagy frusztrálóbb rész, de sikerült rajtuk túllendülni, megoldani a felmerülő problémákat. Mikrokontrollert sem használtam még soha, ezért arról is rengeteg új információt tudtam meg. Dokumentációt sem sokszor írtam még, ebben is sikerült tapasztalatokat szerezni, illetve remélem jó lett azért. Nem mellesleg sikerült egy olyan eszközt készíteni, amit szívesen használok majd a projekt munka után is, és nem csak bekerül majd egy fiókba, hogy majd egy lomtalanításkor kidobjam. Még jobb benne, hogy nagyjából csak kidobásra ítélt, elfelejtett, vagy más célra már használhatatlan dolgokból készült valami új. Csináltam egy rövid videót is, sajnos az asztalom limitált méretei miatt kormány helyett billentyűzettel, de a funkciókat így is tökéletesen bemutatja.