

Témalabor – autó lámpa

Feladat ismertetése:

A feladatunk egy olyan lámpa magasság állító elkészítése volt, ami dinamikusan leköveti a model autó hasmagasságát. A felhasznált eszközök egy arduino, egy proximity szenzor (ami az autó aljának távolságát méri a földtől) és egy servo motor, ami állítani képes a lámpák vízszintessel bezárt szögét. Feladat része volt egy labview „kezelőfelület” létrehozása is.

Feladatmegoldás – arduino: C - setup

Elsőnek felkapcsoltuk a lámpákat a kocsi elején, amik a 5 és 10-es pineken elérhető, majd az enkóder-t állítottuk be, hogy mérjen.

A kar kiinduló állapotának a legfelső állapotát választottuk, így a 27-es regiszterre HIGH és a 26-os regiszterre LOW jelet adunk, hogy a karok elkezdjenek felfele menni.

Mivel a fényviszonyoktól, az autó állapotától (a gumi nem mindig húzza vissza a legfelső pozícióba) és a felülettől függ, hogy mennyit mér a szenzor úgy döntöttük, hogy legelsőnek feltérképezzük az autó kezdő helyzetét. Ezt 100 iteráción át mért távolságadatokat kiátlagolásán keresztül értük el.

Tehát a következő feladat a szenzorból adat kinyerése a 13-as (hex) címen keresztül. Ahhoz, hogy használni tudjuk a szenzort, először kapcsolatot kellett vele létesíteni (beginTransmission) I2C-t használva. Utána a 80-as (hex) címre írással kiválasztottuk a parancs regisztert és a FF-re (hex) írással engedélyeztük a távolság mérést. Hasonlóképpen a mérési sebességet is beállítottuk. A 82-es regiszterre írással kiválasztottuk a távolságmérés gyorsaságáért felelős regisztert és 00 regiszterre írással beállítottuk azt 1.95 mérés / távolságra.

Ezután elkezdhattuk a távolság lekérdezést, amit 100x tettünk meg, miközben az eredményt átlagoltuk.

A ciklusba egy enyhe késleltetést tettünk, hogy a karok a ciklus végére biztosan elérjék a kezdő pozíciójukat. A ciklus végén a motor mozgását leállítottuk.

Több kísérlet alapján megmértük, hogy a kar legfelső és legalsó értéke között az encoder által állított számlálónál nagyból 15.000 eltérés van. Az irányváltás vagy megállás parancskor fellépő késleltetés miatt a motor, így a számláló is picit túlpörög az előre beállított értékekhez képest. A gyakorlatban minimum 30-50 túlpörgést tapasztaltunk, ami a mostani mozgató logikánk használata közben minden esetben a negatív tartományba vitte a számlálót, amikor a számlálót 0 alapértékkel indítottuk el.

Mivel negatív számok kezelése bonyolította volna a logikát jobbnak gondoltuk magasabb alapértékről indítani a számlálót, hogy soha se lehessen negatív az értéke, ezért 1000-re állítottuk a számlálót a setup végén.

Feladatmegoldás – arduino: C - loop

A loopban folyamatosan leolvassuk a szenzor által megadott távolság értéket a fentebb említett módon, és annak függvényében eldöntjük, hogy fel, le vagy nem kell mozgatni a motort.

A mozgatás igényét úgy határozzuk meg, hogy a távolság értékeket leképeztük és átskáláztuk az encoder számláló értékeire.

Legelsőnek a mért adatból kivonjuk a setup-ban kiszámolt átlagolt alapértéket, hogy egy 0-tól induló távolság értéket kapjunk, majd ezt megszoroztuk 5.172-vel, amivel átskáláztuk a számláló intervallumára. Ezen felül hozzáadtunk 1000-ret, hogy ugyan arról az értékről induljon, mint a számláló.

A kovertálásból, mérésből és késleltetésből eredő pontatlanságok miatt szükséges volt egy 1000 tűrést használni. Tehát akkor áll meg a kar, amikor a számláló kisebb, mint az átkonvertált távolság + 500 és nagyobb, mint az átkonvertált távolság -500.

A fel és lemozgatásoknál hasonlóan 1000 tűrést használtunk, de még arra is kellett figyelnünk, hogy a motor ne menjen túl a kijelölt 15 ezres intervallumon, ezért a lefele és felfele mozgatásnál is vizsgáljuk, hogy értelmes tartományban van-e az enkóder értéke, mivel ha túlságosan túlcsordulna lehetséges lenne, hogy egy végtelen ciklusba kerül a motor, amivel nem elromlana a rendszer működése.

A serial portra elküldjük az átkonvertált távolság értéket, hogy majd labview kezelőfelületen ábrázolni tudjuk.

Feladatmegoldás – Labview

Innen nem akarunk semmit irányítani, mivel a rendszer autonóm kell annyira legyen, hogy ne kelljen manuálisan felülvizsgálni. A labview timerje default megegyezik az arduino 9600-as baud ratekével, így azt nem kellett piszkálni, csak az arduinon által küldött értékeket kell kiírni, amit ábráztunk Waveform Chart-on és String Indicator-on a program futása közben. A program végén pedig összefoglalásnak Waveform Graph-el is ábráztuk az adatokat.