

Témalaboratórium

Dokumentáció

Gyulai László (S225H3)

Radványi Patrik Tamás (I7A1T1)

1. Feladat

Egy plexicsőben egy pingponglabda került elhelyezésre, melyet a cső alján található ventilátor képes felfelé fújni. Lézeres távolságmérő segítségével állítson elő olyan fordulatszámot, mely a pingponglabdát egy megadott magasságban tartja. A szabályozó programját az Arduino egységen valósítsa meg. A távolság egy LabVIEW kezelői felületen legyen beállítható, ugyanezen a felületen a beavatkozó jelet is mutassa meg.

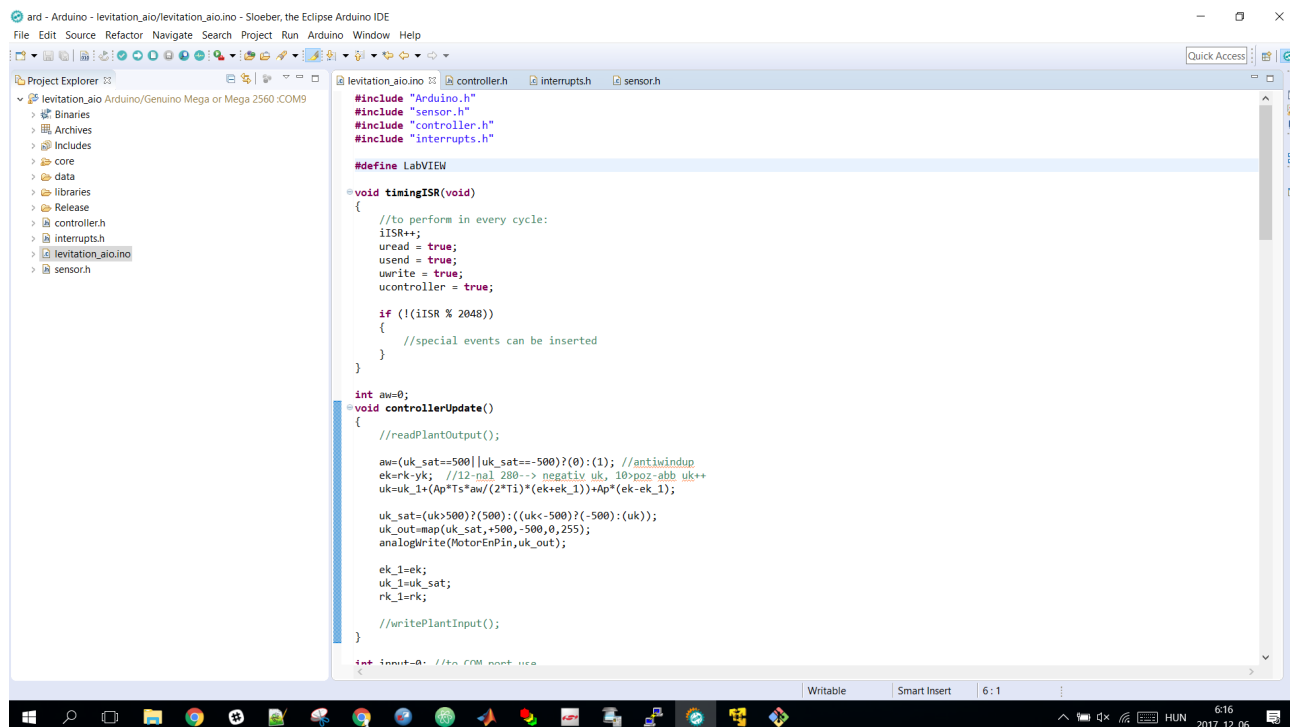
2. Folyamat szabályzolása

A feladat megoldására egy egyszerű PI szabályzót választottunk. Az Arduino, a cső tetején elhelyezett szenzor segítségével I²C protokollon keresztül, a cső tetejétől mért távolságot mintavételezi $T_s = 50$ ms-os időközönként. A mintavételezést (és beavatkozást) a TimerOne időzítő megszakításai kezdeményezik. A szabályzó paraméterei (T_I , A_p) A beavatkozójel számítására az alábbi diszkrét idejű egyenletet használtuk:

$$u_k = u_{k-1} + \frac{T_s A_p}{2T_I} (e_k + e_{k-1}) + A_p (e_k - e_{k-1})$$

Ahol az egyes hibajelek a referencia és a mért kimenet különbségeként keletkeztek. A paraméterek finomhangolását a szakasz beavatkozásra történő reakciójából levont következtetésekkel végeztük. Az implementáció során a beavatkozójel szaturálásával, majd antiwindup beiktatásával csökkentettük az integrátor „túlintegrálását”, illetve offset beállításával kitöltési tényezőre (0..255 tartományba) konvertáltuk. A HMI-vel (LabVIEW) történő kommunikációt soros porton valósítottuk meg. A feladatkírásnak megfelelően olvassuk a referenciajelet, azonban a beavatkozó jel mellett az aktuális

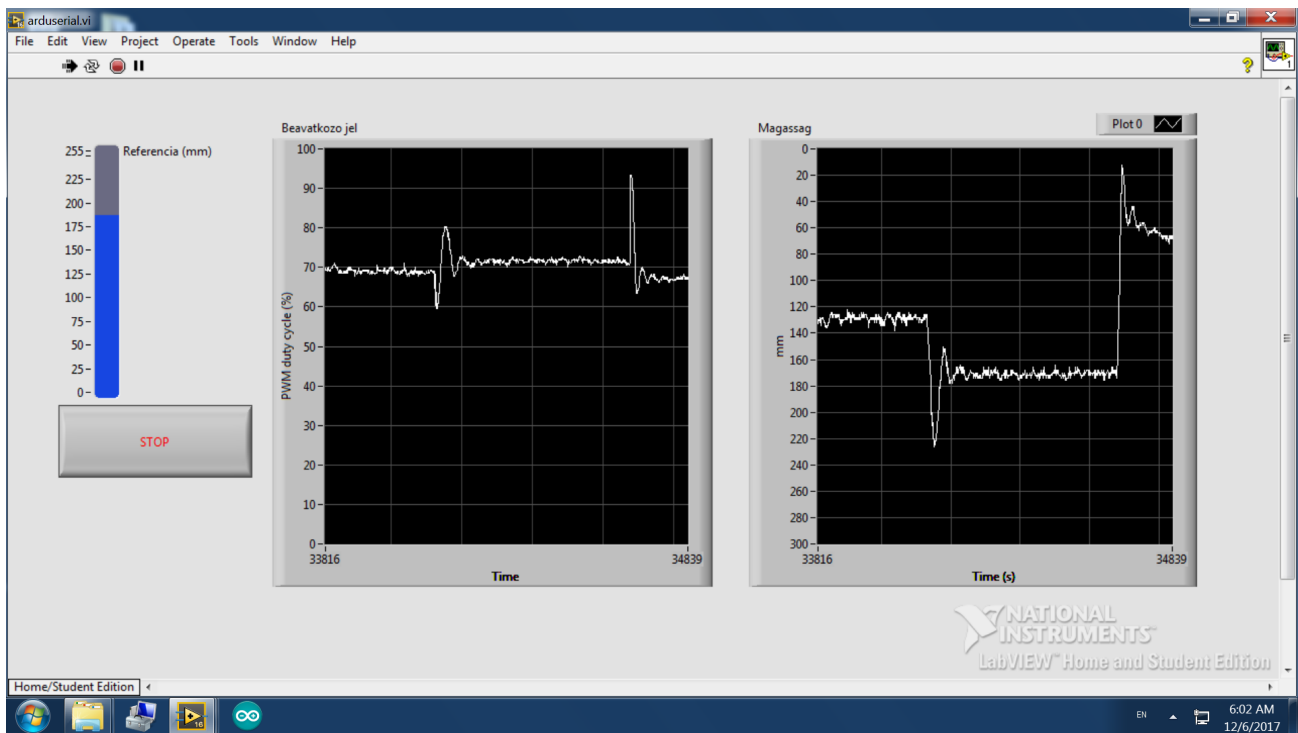
mért magasságinformációkat és számított referenciát is küldjük a PC felé vesszővel elválasztott és pontosvesszővel terminált formában.



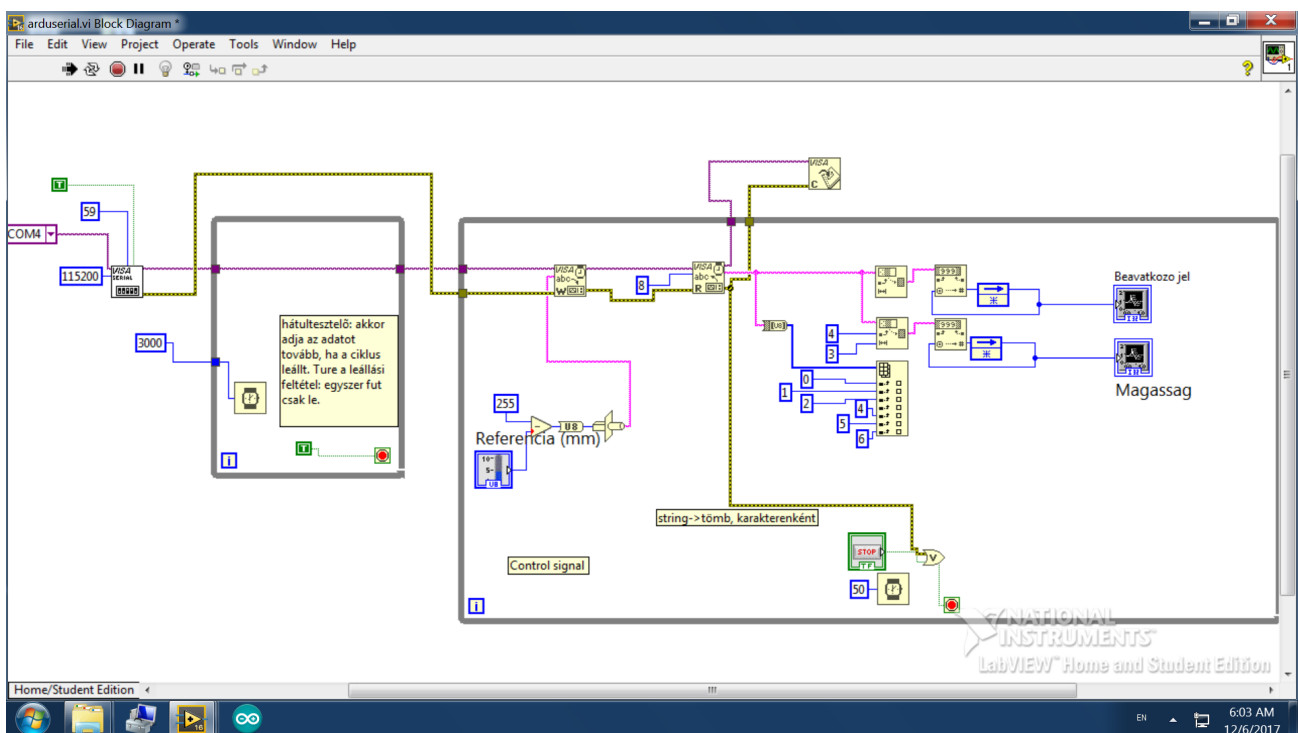
1. ábra. A szabályzó implementálása Eclipse alapú fejlesztőkörnyezetben

3. HMI megvalósítása

A HMI megvalósítása a feladatkírásnak megfelelően LabVIEW programmal készült. A VISA könyvtár segítségével megnyitottuk a soros portot és végtelen ciklusban (STOP gomb megnyomásáig) írjuk a csúszkának értékét (melyet az Arduino referenciaként értelmez) és olvassuk a mikrokontroller által küldött számhármassokat, a beavatkozó jelet és az aktuális magasságot egy-egy diagramon ábrázolva. A skálázást (adattranszformációt) a HMI végzi megjelenés, illetve küldés előtt.



2. ábra. A kezelőfelület felhasználói oldala



3. ábra. A kezelőfelület programozása