# **Ներկառուցված ծրագրի մասնահատկությունները**

Սնուցումը միացնելուց հետո կատարվում են հետևյալ նախնական կարգաբերման քայլերը․

MX\_GPIO\_Init(); // Լռելայն արգելակը միացած պետք է լինի։

MX\_TIM3\_Init();

MX\_TIM4\_Init();

MX\_UART5\_Init();

MX\_I2C1\_Init();

MX\_DAC\_Init();

LCD\_Init();

// ?? Keypad\_Init();

~~Սնուցման հետ միաժամանակ միանում է արգելակի համակարգը (normal close relay) և ֆիքսում սանրը։ Հետ է քաշվում նաև բլոկավորող համակարգը` ազատելով ձեռքի պտուտակը:~~

Այս քայլերից հետո ծրագիրը հատուկ ռեգիստրից (սնուցումից անկախ) կարդում է (Read\_Coord()) սանրի փաստացի կոորդինատը(այսինքն թղթի կտրվելիք չափսը) և արտացոլում է (LCD\_Write()) ցուցասարքի վրա։ Ցուցասարքը արտացոլում է նաև ցանկալի կոորդինատը (որտեղի՞ց է վերցնում տվյալը)։ Քանի դեռ օգտագործողը չի փոխել սանրի կոորդինատը, ապա ցանկալի արժեքը կլինի նույնը, ինչ փաստացի արժեքն է։

Օրինակ․

Real 0200.00 mm – փաստացի կոորդինատ

Set 0200.00 mm – ցանկալի կոորդինատ

~~Եթե օգտագործողին բավարարում է թղթի կտրվելիք չափսը, կարող է անցնել կտրելու փուլ, հակառակ դեպքում ստեղնաշարի օգնությամբ կարող է մուտքագրել նոր կոորդինատը և հրամայել սանրին գնալ նոր կոորդինատ~~։

Սանրին հրամայելու համար․

Ինչ որ բան մեզ հուշում է ? , որ Ծրագիրը սպասում է ստեղնաշարից թվի մուտքագրմանը:

1. Ծրագիրը սպասում է ստեղնաշարից թվի մուտքագրմանը(Read\_Keypad())
2. Եթե մուտքագրվածը թիվ է, ապա ծրագիրը պատկերում է ցուցասարքի համապատասխան դիրքում(LCD\_Write())
3. Ծրագիրը պահում է մուտքագրված թիվը վերջնական թիվը որոշելու համար
4. Ծրագիրը կրկնում է 1-3 կետերը, քանի դեռ ստեղնաշարից ստանում է 0-9 թվերը
5. Եթե ծրագիրը ստացել է 0-9-ից կամ #-ից տարբեր նշան, արհամարհում է այն և շարունակում է 1 կետից
6. Եթե ստացել է # նշան, ապա 3 կետում պահված թվանշաններից կազմում է թիվ
7. Բաղդատում է կազմված թիվը ընթացիկ դիրքի հետ
8. Որոշում է շարժման ուղղությունը, մեծությունը

Մեծությունը փաստացի և ցանկալի կոորդինատների տարբերությունն է։

Այսքանով որոշվեց կտրվելիք թղթի չափսը։ Ցուցասարքը կարտացոլի նոր կոորդինատը(ցանկալի կոորդինատը) և փաստացի կոորդինատը։

Օրինակ․

Real 0200.00 mm – փաստացի արժեք

Set 0399.99 mm – ցանկալի արժեք +/- 0.05 ?

Այնուհետև ծրագիրը տեղափոխում է սանրը անհրաժեշտ դիրք՝ կատարելով հետևյալը.

1. Անջատում է սանրի արգելակը (char state Brush\_UnLock(void); )

0 = unlock,

1 = lock;

1. Շարժման ուղղությունն ու մեծությունն որոշելուց հետո միկրոպրոցեսորը շարժման մեծությանը համապատասխան PWM է գեներացնում։ Այս ազդանշանը LM358-ով պատրաստված DAC-ի միջոցով դառնում է 0-10Վ անալոգային ազդանշան։ Հաճախային վերափոխիչը այս ազդանշանով կարգաբերում է սանրի շարժիչի արագությունը ( 1-50Hz)։ Արագության կարգաբերումից հետո միկրոպրոցեսորը միացնում է (Set\_Inverter()) hաճախային վերափոխիչի առաջ(Brush\_Forward) կամ հետ(Brush\_Back) ելքը:
2. Էնկոդերի յուրաքանչյուր քայլին ծրագիրը կարդում է էնկոդերի արժեքը(Read\_Encoder()) սանրի դիրքը պարզելու համար(էնկոդերը միացված է սանրին և գրանցում է սանրի տեղաշարժը), հատուկ(backup) ռեգիստրում գրում է սանրի կոորդինատը(Save\_Coord())
3. Ցանկալի կոորդինատին մոտենալիս դանդաղեցնում է շարժիչի արագությունը(Change\_Speed(ramp-down) (Change\_Speed(ramp-up) շարժումը սկսելիս)) և հասնելուն պես՝ անջատում հաճախային վերափոխիչը (Set\_Inverter()) ու միացնում է սանրի արգելակը(Brush\_Brake\_ON()), որպեսզի շարժիչը իներցիայով առաջ չգնա։

Ցուցասարքը կարտացոլի նոր(փաստացի) կոորդինատը, գրանցում է backup register-ներում և օգտագործողը կարող է անցնել կտրման փուլ։

Օրինակ․

Real 0400.00 mm – փաստացի արժեք

Set 0400.00 mm – ցանկալի արժեք

Կտրելու Փուլ։

1. Կտրելու փուլ անցնելու համար օգտագործողը պետք է սեղմած պահի ոտնակը(պեդալը)։ Աշխատում է մամլիչի հիդրավլիկայի առաջին փուլը: Մամլիչը ֆիքսում է թուղթը։ Այս գործողությունը մեխանիկական է և միկրոպրոցեսորը չի մասնակցում։ Ծրագիրը ստուգում է, եթե սեղմված է պեդալը(Read\_Pedal()), սկիզբ է դնում է 5 վայրկյանանոց հապաղմանը, որի ավարտից հետո ծրագիրը ակտիվացնում է կտրելու կոճակները (Cutting\_Buttons): Հապաղումից հետո, եթե ոտնակը դեռ սեղմված է, ապա ծրագիրը միացնում է (Solid\_On()) Solid\_N1-ը (Pedal\_Out), որի կոնտակտները սնում են Solid\_N3 -ին(Cutting ) և Solid\_N4-ին(Press\_Again), որոնք նախատեսված են թուղթը ավելի ուժգին սեղմելու և կտրելու համար։
2. Միաժամանակ սեղմած պահել կտրելու համար նախատեսված երկու կոճակները(Cutting\_Buttons)։ Ծրագիրը ակտիվացնում է (Pressing\_On()) հիդրավլիկի փականը(Press\_Again), որը սկիզբ է դնում 3 վայրկյանանոց հապաղմանը։ Մամլիչը էլ ավելի ուժգին է սեղմում թուղթը։ Հապաղումից հետո ծրագիրը ակտիվացնում է (Cutting\_On()) դանակի կցորդիչը(Cutting): Դանակը իջնում, կտրում է թուղթը և վերադառնում է ելման դիրք։ Դանակի թափանիվի վրա տեղակայված երկու տվիչների արժեքները կարդալով(Read\_Knife\_Sensors()) ծրագիրը որոշում է երբ ակտիվացնի և երբ անջատի դանակի կցորդիչը։ Դանակի կցորդիչը անջատելուց (Cutting\_Off()) հետո, ծրագիրը անջատում է հիդրավլիկայի փականը(Pressing\_Off()): Որից հետո օգտագործողը բաց է թողնում ոտնակը և մամլիչը ազատում է կտրված թուղթը։
3. Ակտիվանում է սանրի տեղաշարժելու մեխանիզմը և կտրված թուղթը հրում է առաջ, որպեսզի օպերատորը հեշտ աշխատի։

Թուղթը նորից կտրելու համար պետք է սկսել 1-ին կետից։

Օգտագործողը կարող է նաև ձեռքով տեղաշարժել սանրը՝ բռնակը սեղմած պահելով։ Ծրագիրը ստուգում է (Read\_Hand\_Catch\_Input()) համապատասխան մուտքը(Hand\_Catch) և եթե նկատվել է բռնակի սեղմում, ապա ծրագիրը անջատում է սանրի արգելակը(Brush\_Brake\_OFF())։ Այնուհետև օգտագործողը կարող է տեղաշարժել սանրը։ Ցանկալի դիրք տանելուց հետո օգտագործողը թողնում է բռնակը և ծրագիրը միացնում է արգելակը(Brush\_Brake\_ON())։ Ընթացիկ կոորդինատը արտացոլվում է ցուցասարքին(LCD\_Write()) և ծրագիրը պահում է սանրի կոորդինատը պահեստային ռեգիստրի մեջ(Save\_Coord())։ Եթե համակարգը անջատված է, ապա օգտագործողը չի կարող տեղաշարժել սանրը, քանի որ այն բլոկավորված է բլոկավորող համակարգի կողմից։

1. State machn
2. Current position forever
3. KeyPad read (10ms interrupt)
   1. algorithm
4. LCD (1ms)
   1. Buffer 1 char, RegSel (WritePtr, RdPtr)
   2. Buffer 2 – set of data to send to I2C
      1. Counter.

16 char

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

0 1 2 3 4 5 6 7

8 9 10 11 12 13 14 15

1000 1001