

TRIMIX INSTRUKCIJOS

Table of Contents

Įvadas.....	3
Valdymas per USB.....	4
Naudojamos komandos:.....	4
ATI – informacija.....	4
ATS – parodyti konstantas ir išmatuota kameros ilgį.....	4
ATThh:mm:ss – nustatyti RTC laiką.....	4
ATDyyy.mm.dd – nustatyti RTC datą.....	4
ATWnn=x.xxxxx – įrašyti konfiguracionę konstantą į EEPROM.....	5
ATB – parodyti EEPROM turinį.....	5
ATZ – Trimix inicializacija (RESET).....	5
AT&F – fabrikinį nustatymų atstatymas.....	5
ATCnn – programos būklės perjungimas.....	5
ATH – sustabdyti parodymų srautą per USB/SERIAL.....	5
ATG – paleisti parodymų srautą per USB/SERIAL (CSV).....	5
ATOxxxx... - perduoti komandą į O2 sensorių.....	6
ATAnn – pakeisti LCD ekrano pašvietimo intensyvumą.....	6
ATV – įrašyti konstantas į EEPROM.....	6
ATUaa=hh – įrašyti baitą į EEPROM.....	6
ATQ – parodyti kvadratinės lygties korekcijos koeficientus.....	6
ATKnn – įvesti mygtuko paspaudimą.....	6
EEPROM.....	7
Intro.....	7
EEPROM memory map.....	7
Klaidų kodai.....	8
Valdymas naudojant mygtukus.....	9
Hardware.....	10
Ekranas.....	10
MCU.....	10
EEPROM.....	10
Maitinimo šaltinis.....	10
RS485 (opcija), RS232 (opcija).....	10
Mygtukai (KEYS).....	10
RTC ir kalendorius.....	11
SD kortelė (opcija).....	11
LED.....	11
Fabrikinis konfigravimas (kalibracija).....	12
Teorinė dalis.....	13
Tiesinės priklausomybės kalibracija.....	13
Trijų taškų kalibracija – kubinis spline, beziel ir t.t.....	13
Algoritmas iš programos.....	13
Einamasis matavimas.....	14
Helio skaičiavimas.....	14
Kvadratinės priklausomybės parametrų skaičiavimas.....	14



Įvadas

Šis įrenginys matuoja deguonies ir helio kiekį sausose dujose. Deguonis matuojamas tiesiogiai naudojant specialų sensorių. Helis matuojamas netiesiogiai, nustatant garso greičio pokyčius žinomo ilgio kameroje.

Įrenginio centras STM32F103 serijos mikrokontroleis dirbantis 72MHz dažniu, turintis ADC, UART, I2C, FSMC, GPIO ir USB prievadus. Didelis darbinis dažnis leidžia be problemų atiduoti duomenis per visus prievadus ir atlikti *float* matematiką.

Prietaisas matuoja „judančias“ dujas. Jei per prietaiso jutiklius nevyksta dujų srautas, dėl helio įsigėrimo į plastiką, galimas helio parodymas. Judančiom dujom jokie įsigėrimo efektai nesvarbūs.

Todėl negalima kalibruoti nulinės helio reikšmės (taruori) su stovinčiu oru.



Valdymas per USB

Pajungus Trimix prie bet kokio kompiuterio, automatiškai įsidedia virtualus nuoseklusis (COM) portas (USB CDC). Jokių draiverių nereikia, jokių parametrų nereikia nustatinėti. Visas valdymas atliekamas per bet kokią terminalinę programą ar net CMD. Svarbu, kad komanda baigtusi CR/LF baitų seką (carriage return, line feed. Tai Windows/Dos standartas. Turi veikti su Unix ir Apple OS). Atsakymai iš įrenginio irgi užbaigiami CRLF seka. Visos komandos rašomos didžiosiomis raidėmis. Visos komandos prasideda „AT“ seka (attention. Tai Hayes standarto palikimas).

Naudojamos komandos:

ATI – informacija.

Atsakymas- programos pavadinimas ir versija. Jei veikia RTC, parodoma įrenginio veikimo data ir laikas.

```
TRIMIX VABOLIS.LT 2025 V1.x  
10:40:53
```

ATS – parodyti konstantas ir išmatuota kameros ilgį

Parodo visas konstantas iš kintamųjų, vieną kartą išmatuoja kameros ilgį ir viską parodo terminale. Pavyzdinis tekstas:

```
Konstantos:  
[1]Lair(1)=80.760028  
[2]Lhe100(2)=10.516824  
[3]x^2 delta=10.856000  
[4]...=0.000000  
[5]ADC*=0.000724  
[6]ADC+=0.000000
```

ATThh:mm:ss – nustatyti RTC laiką

Komanda nustato įrenginio vidinį RTC laikrodį. Hh-valandos(00-23), mm- minutes (00-59), ss- sekundės (00-59). Įvedus neteisingus duomenis kas bus nežinau. Reikia rašyti „leading zero“.

Patogu tiesiai iš komandinės eilutės parašyti: *echo ATT%TIME% >COM2*

Ir jei įrenginys pajungtas prie COM2, hosto (kompiuterio) ir įrenginio RTC sinchronizuojasi.

Komanda be parametrų parodo prietaiso laiką.

ATDyyy.mm.dd – nustatyti RTC datą

Pas STM32F103 seriją ši komanda beprasmė, nes ši mikroschema neturi nepriklausomo kalendoriaus. Komanda be parametrų parodo prietaiso data.



ATWnn=x.xxxxx – įrašyti konfiguracionę konstantą į EEPROM

Čia nn yra konstantos numeris (šešiolyktainėje sistemoje. Šiame projekte manau bus mažiau, tai tiks iš dešimtainė). Leading zero privalomas. Po lygybės ženklo eina *float* skaičius. Jei skaičius neigiamas, reikia rašyti su minusu. Pvz ATW01=-3.1415926. Po sėkmingo įrašymo, atnaujinama CRC32 kontrolinė suma ir nauji visi parametrai nuskaitomi iš EEPROM (kaip po ATZ komandos). Kvadratinės kreivės koeficientai perskaičiuojami.

Pavyzdinis tekstas po komandos “ATW01=320.001”:

```
Write to EEPROM @01=320.001000
Wrote settings to EEPROM.
CRC32:0332A38A ROMCRC:0332A38A
```

ATB – parodyti EEPROM turinį

Parodo pirmus 128 EEPROM baitus ir CRC32. Pavyzdinis tekstas po komandos:

```
EEPROM FOUND
AA 55 4C 0D 0B 01 40 00 BC 74 93 18 04 00 74 40
00 00 00 00 00 80 86 40 00 00 00 00 00 00 54 40
EA 41 41 29 5A B9 47 3F 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 10 BE
CE 00 00 00 00 00 00 00 01 4D 33 20 36 38 4C 36
35 32 33 43 55 53 2D 43 43 43 20 53 43 05 45 03
0D 49 92 00 48 41 4A 35 39 33 45 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 98 8A A3 32 03
CRC32:0332A38A
```

ATZ – Trimix inicializacija (RESET)

Po šios komandos nuskaitoma konfigūracija iš EEPROM ir paleidžiamas analizatorius. Kvadratinės lygties koeficientai perskaičiuojami.

AT&F – fabrikinį nustatymų atstatymas

Į EEPROM įrašomi hardcoded fabrikiniai nustatymai. Visos konfigūracijos prarandamos. Naudojamas pajungus naują EEPROMĄ arba jei prikonfigūruota visokių nesąmonių.

ATCnn – programos būklės perjungimas

Tai „mode“ mygtuko paspaudimo analogas. Iškart galima perjungti į norimą programos būklę. Nulinė būklė- pagrindinis ekranas, atsistato po timeout (120 sekundžių). Leading zero privalomas.

ATH – sustabdyti parodymų srautą per USB/SERIAL

Po šios komandos per konsolę neberodami aktualūs parodymai. Matavimai tebevyksta. Po perkrovimo teksto rodymas sustabdomas.

ATG – paleisti parodymų srautą per USB/SERIAL (CSV)

Po šios komandos per konsolę rodami visi matavimai. Pavyzdinis tekstas po komandos:



TRIMIX FACTORY MANUAL

Normal Run mode

TIME, OXIGEN, GAS_TEMP, He, LENGHT

19:31:36, 21.175, 25.906, -0.599, 81.3131, 1.20000, 22.8000

19:31:37, 21.214, 25.937, -0.362, 81.1987, 1.20000, 22.8000

19:31:38, 21.179, 25.937, -0.792, 81.4065, 1.20000, 23.0000

19:31:39, 21.225, 25.937, -0.830, 81.4246, 1.20079, 23.2000

...

Čia buvo normalaus darbo režimas. Kiti režimai taip pat duoda duomenų srautus.

ATOxxxx... - perduoti komandą į O2 sensorių

Ši komanda perduoda tekstinę (xxxx) komandą tiesiai į O2 sensorių. Komandos netikrinamos. Komandos aprašytos O2 sensoriaus instrukcijoje.

ATAnn – pakeisti LCD ekrano pašvietimo intensyvumą

Čia nn yra šešiolyktainis skaičius (00-FF). Šis parametras tiesiogiai perduodamas į PWM generatorių kuris sujungtas su LCD ekrano pašvietimu. Kartu ši reikšmė įrašoma į EEPROM atmintį, į nulį adresą.

Kartais naudojamas patikrinti programos darbui jei neveikia ekranas.

ATV – įrašyti konstantas į EEPROM

Įrašo visas konstantas į EEPROM. Po kamandos pamatom:

Wrote settings to EEPROM.

CRC32:C100D5F6 ROMCRC:C100D5F6

ATUaa=hh – įrašyti baitą į EEPROM

Įrašo duotą baitą į pirmus 256 atminties adresus (galima sugadinti kalibraciją, jei neteisingai naudoti). Adresas ir baitas šešiolyktainėje sistemoje. Komanda: ATU01=66 įrašo į EEPROM 0x66 vertę į 0x01 adresą. CRC32 atnaujimama automatiškai. Po kamandos pamatom:

Wrote settings to EEPROM.

CRC32:C100D5F6 ROMCRC:C100D5F6

Kad peržiūrėti ką įrašėm, komanda ATB.

ATQ – parodyti kvadratinės lygties korekcijos koeficientus

Atsakymo pavyzdys:

a=-0.0088 b=-0.6203 c=107.4971

ATKnn – įvesti mygtuko paspaudimą

Įvesti baitą nn į mygtuko registrą. Fiziniai mygtukai tai bitai iš eilės, pradedant LSB. Šešiolyktainėje sistemoje.



EEPROM

Intro

Šiame įrenginyje naujama standartinė I2C EEPROM mikroschema. Prototipe- iš DRR SPD modulio. Ši mikroschema parinkta dėl beveik neriboto perrašymų kiekio ir paprasto naudojimo. Naudojam tik pirmus 128 baitus (kol kas). Paskutiniai 4 baitai (0x7C – 0x7F) naudojami kaip atminties CRC32 saugykla. Jei CRC32 iš paskutinių baitų nesutampa su likusios atminties paskaičiuota CRC32 pranešama apie atminties klaidą.

EEPROM turinys nuskaitomas įjungus prietaisą arba davus *init/reset* komandą

Pirmi 8 baitai rezervuoti. Toliau eina *double* formate programos konstantos. Kiekviena konstanta užima 64 bitus (8 baitus).

EEPROM memory map

#0: 0x00 – 0x07 – baitinė atmintis:

00:	LCD LED PWM. Default: A0
01:	rezervuota
02:	rezervuota
03:	rezervuota
04:	rezervuota
05:	rezervuota
06:	rezervuota
07:	rezervuota

#1: 0x08 – 0x0F – double konstanta: ox_b1, kameros ilgis ore, default: 81.8

#2: 0x10 – 0x07 – double konstanta: ox_b2, kameros ilgis 100% He, default: 13.4

#3: 0x08 – 0x0F – double konstanta: ox_b3, kvadratinė nuokrypa, default: 10.856

#4: 0x20 – 0x07 – double konstanta: ox_b4, rezervuota

#5: 0x08 – 0x0F – double konstanta: ox_adc1, ADC daugiklis, default: 0.000724

#6: 0x30 – 0x07 – double konstanta: ox_adc2, ADC poslinkis, default: 0.0

... rezervuota

0x7C – 0x7F - CRC32



Klaidų kodai

Visos klaidos ir programos būsena saugoma viename 32 bitų kintamajame: HSTATUS. Bitų vienetas rodo, kad klaida aktyvi, nulis- klaidos ar būsenos nėra. Visi bitai lentelėje:

Bitas	Aprašymas
0	EEPROM FAIL. Neaptikta EEPROM mikroschema, I2C buss klaida.
1	CRC32 FAIL. Kontrolinės sumos nesutampa- atminties klaida, konfigūracijos klaida.
2	O2 FAIL. O2 jutiklis išdavė klaidos pranešimą.
3	CALC OVERFLOW. Dėl neteisingų parametrų skaičiavimuose gavosi klaida.
4	TEMP OUT OF RANGE. Temperatūra išėjo iš jutiklių darbo režimų.
5	ADC GENERIC ERROR. Reiškia kad arba užstrigo softas arba sudegė MCU.
6	NO CALIBRATION, FACTORY SETTINGS. Prietaisas nesukalibruotas.
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	
29	
30	
31	INIT DONE. Prietaisas praėjo visus testus. Normalus darbas.

Jei prietaiso būklė yra 0 arba -1 (0xFFFFFFFF) vadinasi įvyko totali klaida.



Valdymas naudojant mygtukus

Šiuo metu numatyti du mygtukai (Key1) ir (Key2). Key1 perjungia darbo režimą (mode). Key2 naudojamas patvirtinti komandai (Enter). Key4 – cancel.



Hardware

Ekranas

Šiuo metu naudojamas darbinis grafinis LCD ekranas su baisiu mėlynu pašvietimu iš Husqvarna žoliapjovės. Programa parašyta taip, kad teoriškai galima pajungti bet koki ekraną. Joks dizainas šiuo metu nenaudojamas. Visa informacija rodoma tekstinėmis eilutėmis. Ekraną apačioje yra būklės juosta (status). Joje rodoma judanti širdelės animacija- programos 1s taimerio ir pagrindinio ciklo indikacija ir HSTATUS kintamojo šešiolyktainė reikšmė.

MCU

STM32F103VGT. Teoriškai bet koks iš STM32 serijos mikroprocesorių kurie turi reikiamas elementus: GPIO, USB, I2C, ADC ir SPI/FSMC/DMA/INT ekranui. MCU naudoja kelių lygių pertraukimus ir DMA: dėl deguonies sensoriaus asinchroninio darbo, ekranas dirba dalinai per DMA ir INT, USB naudoja INT, mygtukai gali naudoti INT (šiuo metu nenaudoja).

EEPROM

Bet kokia I2C EEPROM mikroschema. Jei naudojama su kitoku adresu, pakeisti kode „#define I2CR 0b1010000“ į tinkamą. Šiuo metu panaudota mikroschema iš DDR SIM modulių, greičiausiai- M34E02, gamintojas tas pats kaip ir MCU, STMicroelectronics.

Maitinimo šaltinis

Šiuo metu (prototipe) naudojami 7805 čipas pagaminti 5V maitinimą (jei maitinti iš išorinio maitinimo šaltinio). Dėmesio! Tai tik testavimo maitinimo šaltinis- labai mažas radiatorius. Pagrindinis prietaiso maitinimas- klasikinis USB (USB mini). Energijos sunaudojimas apie 2W su prototipo su LCD ekranu.

Atstumo sensoriumi reikia virš 15V maitinimo šaltinio. Šiuo metu panaudotas kitiškas step-up modulis iš aliexpreso. Ateityje reikėtų pagaminti „custom“ maitinimo šaltinį arba panaudoti profesionalius modulius.

RS485 (opcija), RS232 (opcija)

Visą informaciją galima perduoti ne tik per USB CDC, bet ir per UART. Šiuo metu, prototipe yra sumontuotas industrinis RS485. Keičiant schemą galima padaryti RS232.

Mygtukai (KEYS)

Šiuo metu prietaise yra keturi mygtukai. KEY1 – programos režimo pasirinkimas, KEY2 – OK, KEY4 – Cancel.



KEY3 – kol kas neturi panaudojimo.

RTC ir kalendorius

Esama firmwarė nenaudoja RTC (real time clock) ir kalendoriaus savo algoritme. Vienintelis panaudojimas- 1s darbo ciklas. Jį galima suorganizuoti ir su vidiniu timeriu. RTC pas STM32 mikroschemas beveik visada yra. Jei reikia išlaikyti laiką išjungus, reikalinga baterija arba jonistorius (supercap). Taip pat, pageidautinai 32kHz laikrodis kvarcas.

STM32F103 serijos mikroschemos neturi aparatinio kalendoriaus. Todėl kalendorius šiame projekte beprasmis. Naudojant naujesnės kartos mikroschemas, kalendorius galimas.

SD kortelė (opcija)

Naudojamas MCU leidžia pajungti SD kortelę ir joje kažką rašyti. Šiuo metu ši opcija net nesuprogramuota.

LED

Naudojamas MCU leidžia pajungti iki keliasdešimt LEDų indikacijai. Šiame projekte kol kas jokių LED nėra. (Neskaitant LCD ekrano pašvietimui).



Fabrikinis konfigravimas (kalibracija)

Pasiūloma semi auto kalibracijos procedūra:

1. Paduodam 10V į atsumo jutiklio pajungimo lizdą. Pasitikrinam, kad ant MCU analoginio įėjimo yra kiek mažiau nei 1.2V. (viršijus ADC max įtampa galima sudeginti MCU ADC).
2. Pajungiam atstumo jutiklį. Paleidžiam orą be He. Palaukiam parodymų stabilizacijos.
3. Sukalibruojam kameros ilgį, kai oras.
4. Paleidžiam 100% helį. Palaukiam parodymų stabilizacijos.
5. Sukalibruojam kameros ilgį, kai 100% He.
6. Perjungiam į ilgio matavimo režimą.
7. Leidžiam žinomus dujų mišinius, išmatuojam ilgį. Gaunam vieną ar du papildamus kalibracijos taškus. Paskaičiuojam tiesinę ir parabolinę (kvadratinę) kreives.
8. Paskaičiuojam skirtumą tarp tiesinės ir kvadratinės kreivės prie 50% He.
9. Šį skirtumą suvedam į prietaiso EEPROM.

Dėmesio! Dėl helio prasiskverbimo į plastiką, negalima kalibruoti 0% (taruoti) su stovinčiom dujom.



Teorinė dalis

Tiesinės priklausomybės kalibracija

Prietaisas dirbą matuodamas kameros ilgį esant skirtingom dujom. Skirtingos dujos įtakoja garso greitį, ko pasekoje atstumo jutiklis rodo iškreiptus parodymus. Helis dar įtakoja įvairius akustinius fenomenus ir elektronikos bei pjezo elementų veikimą, todėl išmatuotas atstumas dažnai neatitinka teorinių parodymų.

Programa kalibruojasi nekreiptama dėmesio į fizikos teoriją. Ji veikia pagal matematikos dėsnius.

Šiuo metu veikia tiesinė priklausomybė- kalibracija per du taškus.

Skaiciai: ox_b1 , $x1$ – kameros ilgis prie 0% helio, ox_b2 , $x2$ – kameros ilgis prie 100% helio.
Atitinkamai: $y1$ -0%, $y2$ -100%.

Išvedam tiesę per du taškus:

Pokrypis (slope):

$$m = \frac{y_2 - y_1}{x_2 - x_1} = \frac{0 - 100}{x_2 - x_1} = \frac{100}{x_2 - x_1}$$

Atitraukimas (offset):

$$b = y_1 - m * x_1$$

Helio paskaičiavimo formulė, kai žinomas atstumas (x):

$$y = m * x + b$$

Trijų taškų kalibracija – kūbinis spline, bezier ir t.t.

Darant fabrikinę kalibraciją išmatuojami kelis dujų pavyzdžiai su žinomom vertėm ir užfiksuojamas ilgis.

xxx

Algoritmas iš programos

Šis algoritmas (source code) yra programoje.



Einamasis matavimas

```
adc_calibrate();
a=median_filter(adc_read_v_blocking(ADC_CHANNEL_0));
HAL_Delay(1);
adc_calibrate();
a=median_filter(adc_read_v_blocking(ADC_CHANNEL_0));
HAL_Delay(1);
adc_calibrate();
a=median_filter(adc_read_v_blocking(ADC_CHANNEL_0));
atstumas = a * ox_adc1+ox_adc2;
```

ADC kalibruojamas prieš kiekvieną skaitymą. ADC išmatuoja atstumą tris kartus ir kartu su 2 matavimais iš anksčiau, paskaičiuojam „median“ kameros ilgį. Median filtras iš 5 elementų.

Gal daryti ADC skaičiavimus kas 1/4 sekundės, nepriklausomai nuo matavimo ciklo (1s)?

Helio skaičiavimas

```
helis=atstumas*atstumas*coef_a+atstumas*coef_b+coef_c;
```

Helis skaičiuojamas iš atstumo naudojant kvadratinę priklausomybę.

Kvadratinės priklausomybės parametrų skaičiavimas

Dėl prietaiso jutiklių iš šiaip fizikos, ilgio ir helio santykis nėra tiesiškas. Todėl skaičiuojama naudojant kvadratinę priklausomybę. Išmatuojami du kraštiniai taškai (0 ir 100% He) ir keli taškai (vienas taškas arčiau 50% He). Kvadratinis nuokrypis paskaičiuojamas išoriškai ir jo reikšmė prie 50% įvedama į parametrus. Toliau viskas vyksta automatiškai:

```
//TIESINE LYGTIS
m=(100-0)/(ox_b2-ox_b1);
b=0-m*ox_b1;

//Nauja x50 koordinate parabojeje ir nauja reiksme prie 50% He.
x50=-((ox_b2-ox_b1)/2-ox_b2);
y50=m*x50+b+ox_b3;

//suskaiciuojam kvadratine priklausomybe per 3 taskus.
POINTS3(ox_b1,0.0,
        x50,y50,
        ox_b2, 100.0,
        &coef_a, &coef_b, &coef_c
);

// Print_Koef(tmp_txt);
// y50=x50*x50*coef_a+x50*coef_b+coef_c; //pasitikrinam x50 reiksme.
```

