

Cuprins

1	Stadiul actual	2
1.1	Cuptorul electric.....	2
1.2	Termocuplu	3
1.3	Regulatorul automat	4
2	Fundamentare teoretică	6
2.1	Algoritmi de reglare	6
2.1.1	Algorimul de reglare tip P (proporțional).....	6
2.1.2	Algorimul de reglare tip PI (proporțional-integral).....	7
2.1.3	Algorimul de reglare tip PD (proporțional-derivativ)	9
2.1.4	Algorimul de reglare tip PID (proporțional-integral-derivativ)	11
2.2	Reglarea regimului termic la cuptoare	14
2.2	Termocuplu tip K	16
3	Implementarea soluției adoptate.....	17
3.1	Ponderea mărimii prescrise la algoritmul PID	17
3.2	Utilizarea algoritmului PID	18
3.3	Triacul (BTA16B)	18
3.4	Termocuplu (MAX6675).....	20
3.5	Arduino Nano	22
3.6	Dispozitiv driver triac izolat optic (MOC 3020)	24
3.7	Optocupler (PC817).....	25
3.8	Modulul LCD 1602 având interfața serială I2C	26
3.9	Redresor (KBPC1010).....	27
3.10	Codul Arduino folosit și explicat.....	28
4	Rezultate experimentale.....	31
4.1	Rolul fiecărei componente	31
4.2	Setatea controllerului PID	32
4.3	Asamblarea circuitului.....	33
4.4	Testarea	35
	Concluzie	37
	Bibliografie.....	38
	Anexe	39

1 Stadiul actual

Dezvoltarea continuă a științei și tehnici impune realizarea unor reglaje superioare la aplicații industriale asigurând performanțe eficiente din ce în ce mai ridicate, siguranța în exploatare a instalațiilor și echipamentelor, prin realizarea unor produse competitive, cu calitate superioară, având costuri cât mai scăzute și cu reducerea consumului de energie. În cadrul proceselor industriale automatizarea trebuie să urmărească să țină sub control și să regleze în permanență niște parametri de bază a procesului tehnologic respectiv temperatura din incintă, umiditatea aerului, presiunea. Datorită posibilităților de utilizare a simplității constructive, pe scară largă în domenii industriale sunt folosite regulatoare de tip PID.

Prezența documentație propune proiectarea unui sistem automat cu reglarea temperaturii la un cuptor electric prin algoritmul PID și utilizând metode ale gradului maximal de stabilitate realizând acordarea parametrilor.

1.1 Cuptorul electric

Cuptoarele cu rezistoare sunt niște dispozitive care prin efectul Joule-Lenz, transformă energia electrică în energie termică această transformare este realizată cu ajutorul rezistoarelor sau încălzitoarelor, formând un cuptor electric cu rezistoare cu încălzire indirectă. Aceste cuptoare cu rezistoare și încălzire indirectă sunt utilizate pentru tratamente termice respectiv deformări la cald, topiri de metale și aliaje.

Temperatura din cuptoare poate fii:

- $\theta_c < 350^\circ\text{C}$ - temperatură joasă;
- $\theta_c = (350^\circ\text{C} \dots 1000^\circ\text{C})$ - temperatură medie;
- $\theta_c > 1000^\circ\text{C}$ - temperatură înaltă;

Construcția cuptorului va ține cond de regimul temperaturii de funcționare. Regimul de lucru a cuptoarelor poate fi:

- Intermitend (periodic) – funcționarea cuprinde toate etapele (ciclu complet), respectiv încărcarea, încălzirea, menținere la temperatură necesară, răcire și descărcarea materialelor din cuptor. Acest regim este folosit la cuptoare tip cameră.
- Continuu – regimul continuu presupune deplăsarea permanentă a pieselor care se încălzesc de la încărcare până la descărcare. Acest regim este folosit la cuptoare tip tunel.

Un cuptor electric industrial de tip cameră care are rezistoare cu încălzire indirectă este format din următoarele elemente:

- carcasa cuptorului este construit din tablă și profile din oțel asigură rezistența mecanică și stabilitatea construcției.
- căptușeala cuptorului realizată din materiale termoizolante, reflectare, asigură stoparea pierderilor termice pe durata încălzirii, a funcționării cuptorului, calitatea procesului tehnologic, fiabilitatea și eficiența instalației. Grosimea stratului de termoizolație (căptușeală) se stabilește în funcție de parametri de temperatură necesari funcționalității cuptorului.
- ușa cuptorului se face din metal cu straturi de termoizolație sau materiale refractare cu grosimi pentru satisfacerea temperaturilor de funcționare. Cuptoarele de capacitate medie sau mare vor avea uși cu acționare mecanizată.
- rezistoare, elemente încălzitoare se execută din materiale conductoare cu rezistivitate electrică mare și coeficient redus de variație a rezistivității cu temperatura. Se amplasează în camera cuptorului sub forma unor sârme sau benzi pe pereți laterali, sub vatră sau pe bolta camerei cuptorului. Fixarea

rezistoarelor se va face cu elemente de susținere din tuburi ceramice, cărămizi, cârlige și bolțuri, din materiale refractare. Diametrul tuburilor ceramice pe care sunt dispuse rezistoarele, sub formă de spirale sau zig-zag, se alege pentru asigurarea rezistenței mecanice.

- dispozitive rezistente la temperaturi înalte, căi de rulare, șine, ghidaje, plăci de vatră, au rolul de a susține, fiind suport pentru încărcătura din camera cuptorului.

1.2 Termocuplu

Termocuplu este un senzor folosit pentru măsurarea temperaturii în toate domeniile. El funcționează datorită efectului Seebeck care creează o diferență de potențial electric datorită diferențe de temperatură. Termocuplurile sunt utilizate în multiple domenii precum în medii științifice, industriale și inginerie. Pot fi folosite într-o multitudine de setări asigurând timpi de răspuns rapizi oferind și capacitatea de stabilire rapidă și precisă a temperaturilor, a intervalelor de temperatură după o configurație stabilă, specifică termocuplului se poate ajunge la temperaturi de peste 270°C ajungând la 2500°C. Circuitul de măsură a unui termocuplu se poate vedea în Figura 1 și modul de lucru este prezentat în [5].

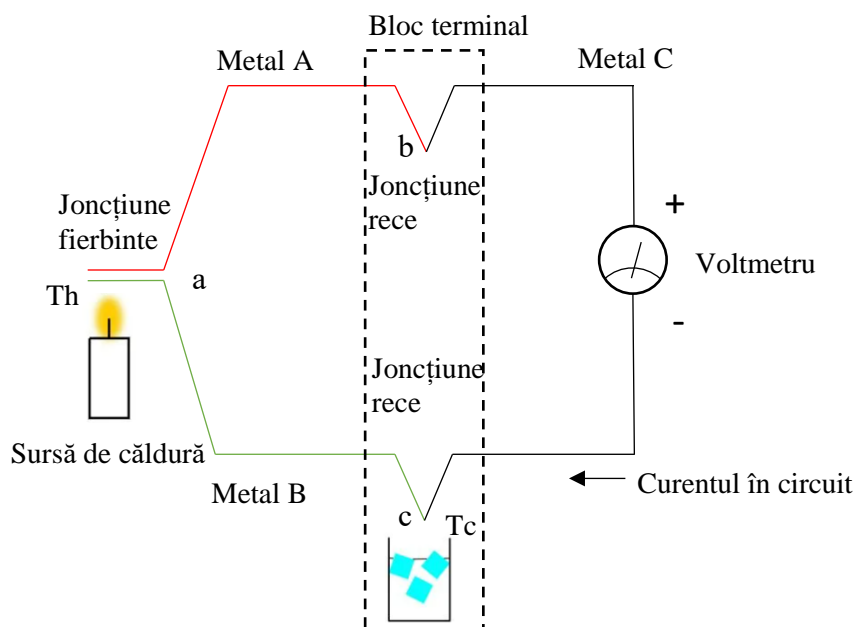


Figura 1. Un circuit de măsură a termocuplului

Termocuplurile sunt instrumente simple robuste și foarte rentabile după tipul lor fiind notate cu litere precum J, K, N, L sau T. În esență termocuplul este format din minim două tipuri de metale cu proprietăți de dilatare diferite care fiind în contact direct formează un circuit cu două joncțiuni electrice separate, una dintre joncțiuni (joncțiunea fierbinte) definește temperatura necunoscută ce trebuie măsurată, iar cealaltă joncțiune (joncțiunea rece) este conectată la corpuri cu temperatura de referință cunoscută.

Diferența de temperatură dintre cele două joncțiuni dă naștere la o tensiune în circuit, fluxul de curent ducând la determinarea temperaturii în punctul de contact “fierbinte”.

Deci termocuplu funcționează când se crează o variație a tensiunii pe circuitul cu materiale din metale diferite, când temperatura se modifică la punctul fierbinte în raport cu punctul rece.

1.3 Regulatorul automat

Regulatorul automat are rolul de a prelua operațional semnalul de eroare, (obținut în urma comparației liniar-aditive a mărimii de intrare r și a mărimii de reacție y_r , în elementul de comparație) și de a elabora la ieșire un semnal de comandă u pentru elementul de execuție.

Cu ajutorul traductorului de reacție se obțin informații curente despre desfășurarea procesului tehnologic, informaționale sunt prelucrate de regulatoare automate după o anumită lege care definește algoritmul de reglare automată. Legile de reglare (algoritmii de reglare) frecvent utilizați în reglarea proceselor tehnologice sunt de tip proporțional-integral-derivativ (PID). O schemă bloc a regulatorului automat se poate vedea în Figura 2 și este prezentat în [3].

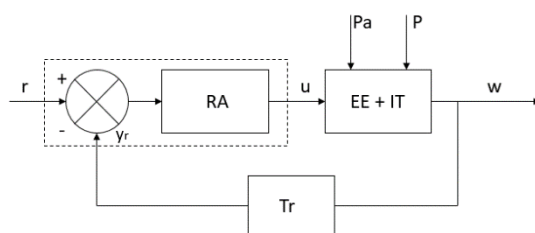


Figura 2. Regulator Automat

Regulatorul electric, pneumatic, hidraulic sau mixt se poate obține prin varietatea construcției regulatorului cu implementarea unor anumiți algoritmi de reglare.

Se pot realiza variante diferite de regulatoare automate, chiar și în cadrul aceleiași categorii constructive, obținându-se anumite avantaje din punct de vedere al preciziei de reglare și a algoritmii de reglare, din punct de vedere al costului, din punct de vedere a siguranței în funcționare, a flexibilității, întreținerii și exploatarei.

Clasificarea regloatoarelor automate este următoarea:

- În funcție de caracteristicile de transfer ale instalației tehnice
 - Controllere cu caracteristici de transfer variate utilizate în următoarele categorii de sisteme:
 - sistem automat de stabilizare ($r = ct.$);
 - sistem de control programatic (se cunoaște $r = f(t)$);
 - sistem de urmărire ($r = f(t)$ necunoscut în avans).
 - Controllere cu caracteristici variabile de transmisie:
 - regulator adaptiv;
 - regulatoare instruibile;
 - regulatoare extremale (optimizatoare automate).
 - În funcție de viteza de reacție a echipamentului tehnic:
 - controller pentru procese rapide;
 - controller pentru procese lente.
 - În funcție de numărul de puteri ale instalației tehnice:
 - regulatoare monovariabile;
 - regulatoare multivariabile.
- Acomodarea proprietăților structurale și funcționale ale regulatorului
 - În funcție de tipul de efect al regulatorului:
 - regulatoare cu acțiune continuă, liniară sau neliniară;

- reglatoare cu acțiuni discrete folosind impulsuri modulate sau numerice.
- După configurarea blocului controlerului și a semnalelor de utilizat:
 - reglatoare unificate;
 - reglatoare specializate.
- În funcție de tipul de suport de informare:
 - reglatoare electronice;
 - reglatoare de presiune a aerului;
 - reglatoare hidraulice;
 - reglatoare mixte (electro-hidraulic, electro-pneumatic).
- În funcție de tipul de reacție utilizat:
 - regulator cu reacție în funcție de variabilele de ieșire;
 - reglatoare care reacționează în funcție de variabile de stare (reglatoare cu estimatoare de stare).

Blocul regulator are mai multe părți componente interconectate funcțional. În Figura 3 este structura blocului regulator de tip industrial și este prezentat în [3].

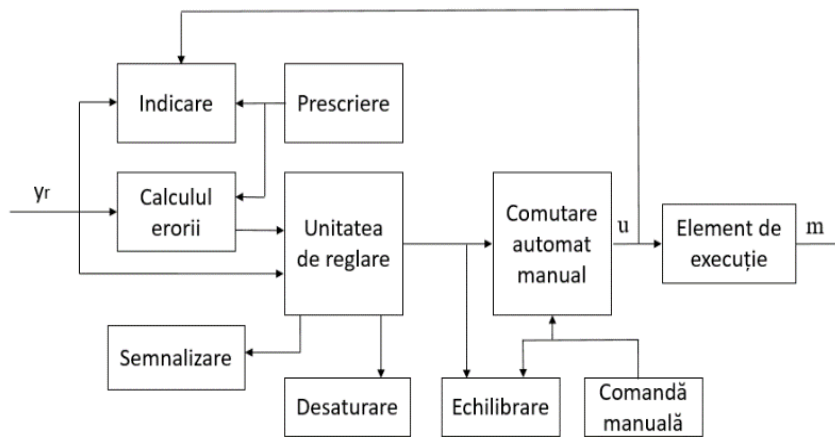


Figura 3. Structura regulatorului automat electric

2 Fundamentare teoretică

2.1 Algoritmi de reglare

În aplicații industriale, în cadrul structurilor cu unul sau mai multe grade de libertate, sunt utilizate legi de reglare respectiv algoritmi de reglare continui sau analogici fiind algoritmi care operează cu mărimi având variația continuă în timp. Vom analiza algoritmi de tip: proporțional (P), proporțional integral (PI), proporțional derivativ (PD) și proporțional integral derivativ (PID).

2.1.1 Algoritm de reglare tip P (proporțional)

Definim algoritmul de reglare de tip P ca fiind cea mai simplă lege de reglare iar într-o structură clasică de SRA cu reacție unitară și un singur grad de libertate având relația (1) prezent în [1].

$$u(t) = K_P \varepsilon(t) \quad (1)$$

Unde : $u(t)$ este mărimea de comandă, $\varepsilon(t)$ eroarea, K_P este factor de proporționalitate și reprezintă parametrul de acord al controllerului de tip P.

Parametrul de acord al controllerului proporțional mai poate fi exprimat sub forma benzii de proporționalitate BP , prin relația (2) prezent în [1].

$$BP[\%] = \frac{100\%}{K_P} \quad (2)$$

Comportarea reală a unui regulator de tip P este determinat de modul de implementare a algoritmului proporțional de reglare, funcția de transfer a elementului de reglaj în domeniul Laplace va fi relația (3) prezent în [1].

$$H_R(s) = \frac{U(s)}{\varepsilon(s)} = K_P \quad (3)$$

Iar dacă la intrarea controllerului vom avea un filtru trece-jos, relația de calcul va avea forma (4) prezent în [1].

$$H_R(s) = \frac{U(s)}{\varepsilon(s)} = K_P * \frac{1}{1+sT_f} \quad (4)$$

unde T_f este constanta de timp al filtrului care limitează banda de frecvență la acțiunea destabilizatoare a zgomotelor de înaltă frecvență.

Ideal, când pentru o treaptă unitară aplicată la intrare, răspunsul va fi o treaptă de amplitudine egală cu factorul de amplificare.

Prin aplicarea unui salt de tip treaptă unitară la intrarea pe canalul erorii se obține răspunsul indicial al controllerului proporțional prevăzut cu filtru, de formă redată în Figura 4.

Alura răspunsului indicial arată o viteză ridicată de răspuns pentru controllerul de tip proporțional. Filtru are un efect de ușoară alterare a frontului crescător pentru semnalul de ieșire. De fapt utilizarea filtrului trece jos impune o viteză finită de modificare a mărimii de comandă $u(t)$ fiind efectul aplicării saltului de tip treaptă pe canalul de eroare $\varepsilon(t) = \sigma(t)$. La determinarea valorii constante de timp a filtrului T_f , folosind răspunsul indicial al controllerului Figura 4 prezent în [1], se procedează astfel:

- Se duce tangentă la curba de răspuns în punctul $t = t_0$.

- Se intersectează tangenta cu o dreaptă orizontală la valoarea constantei de proporționalitate a mărimi de comandă, deci $u(t) = K_P$.
- Prin proiectarea punctului de intersecției pe axa timpului rezultă intervalul $t_1 - t_0 = T_f$, respectiv valoarea constatei de timp.

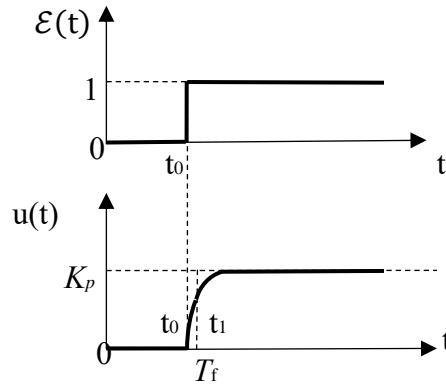


Figura 4. Răspunsul indicial al controllerului proporțional prevăzut cu filtru

2.1.2 Algorimul de reglare tip PI (proporțional-integral)

La componeta de tip proporțională (P) dacă se adaugă o componentă de tip integrală (I) rezultă controllerul proporțional-integral a cărui acțiune caracterizează mărimea de comandă $u(t)$ ca o suprapunere dintre o componentă proporțională cu eroarea $\varepsilon(t)$ și o integrală proporțională cu integrala erorii, exprimându-se prin relația (6) prezent în [1].

$$u(t) = K_P[\varepsilon(t) + \frac{1}{T_i} \int_0^t \varepsilon(\tau) d\tau] \quad (6)$$

unde: K_P este factorul de proporționalitate; T_i este constanta de timp de integrare.

Funcția de transfer a controllerului de tip PI se poate vede în relația (7) prezent în [1].

$$H_R(s) = \frac{U(s)}{\varepsilon(s)} = K_P * (1 + \frac{1}{sT_i}) \quad (7)$$

iar în cazul introducerii unui filtru trece jos pe intrarea controllerului rezultă relația (8) prezent în [1].

$$H_R(s) = \frac{U(s)}{\varepsilon(s)} = K_P * (1 + \frac{1}{sT_i}) \frac{1}{1+sT_f} \quad (8)$$

unde T_f este constanta de timp a filtrului trece jos.

Dacă reprezentăm răspunsul indicial al controllerului PI având filtru la intrare, observăm că mărimea de comandă, în primele momente ale răspunsului are un salt de la zero până la valoarea constantei K_P , ponderat de filtrul trece jos, iar componenta integrală are forma unei rampe ce crește continuu cu panta. Dacă avem o eroare nenulă și persistentă în timp componenta integrală are ca efect saturarea ieșirii elementului de execuție. Răspunsul indicial se poate vedea în Figura 5 prezent în [1].

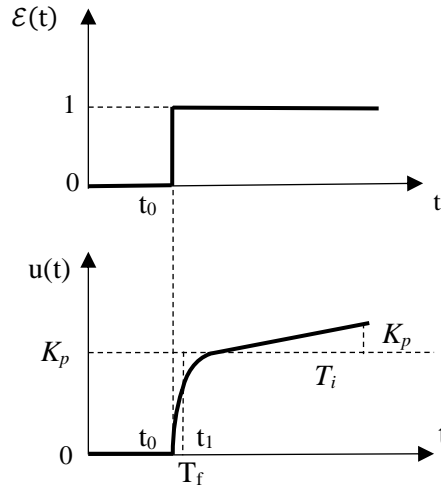


Figura 5. Răspunsul indicial al controllerului proporțional-integral prevăzut cu filtru la intrare

Componenta integrală are rolul de anulare a erorii staționare la poziție a unui controller tip P, definit prin relația (9) prezent în [1].

$$u(t) = K_p \varepsilon(t) + u_b \quad (9)$$

unde $u_b \neq 0$ reprezintă deplasarea comenzii când eroarea a devenit zero. Acest termen, componentă integrală, fiind cauza pentru care controllerul PI nu va avea ieșirea procesului la valoarea impusă prin prescriere, asigurându-se ajustarea automată în regim staționar (eroare zero).

Ajustarea automată a componentei u_b , respectiv introducerea componentei integrale în structura controllerului proporțional, este prezentată în Figura 6 prezent în [1].

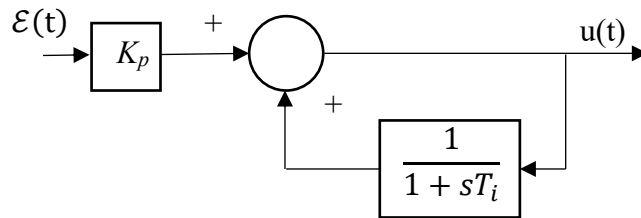


Figura 6. Introducerea componentei integrale în structura controllerului proporțional

Componenta integrală se introduce prin filtrarea ieșirii controllerului, deci se introduce o cale cu reacție pozitivă care obligă controllerul să acționeze și după momentul presupunerii de anulare a erorii, eroarea staționară putând fii adusă la zero, constanta de timp de integrare numindu-se constantă de timp de resetare, iar funcția implementată numindu-se „resetare automată” a controllerului proporțional, exprimată cu relația (10) prezent în [1].

$$U(s) = K_p \varepsilon(t) + \frac{1}{1+sT_i} U(s) \quad (10)$$

de unde configurația unui controller PI cu relația (11) prezent în [1].

$$U(s) = K_P(1 + \frac{1}{sT_i})\mathcal{E}(t) \quad (11)$$

Cu cât constanta de timp este mai redusă cu atât acțiunea de eliminare a erorii staționare este mai rapidă. Totodată caracterul oscilant al răspunsului indicial se accentuează iar T_i tinde la infinit (compensarea erorii staționare se prelungește la timp infinit).

2.1.3 Algorimul de reglare tip PD (proporțional-derivativ)

La componeta de tip proporțională (P) dacă se adaugă o componentă de tip derivativ (D) rezultă controllerul proporțional-derivativ a cărui acțiune caracterizează mărimea de comandă $u(t)$ ca o suprapunere dintre o componentă proporțională cu eroarea $\varepsilon(t)$ și o componentă proporțională cu derivata erorii, exprimându-se prin relația (12) prezent în [1].

$$u(t) = K_P[\mathcal{E}(t) + T_d \frac{d\mathcal{E}(t)}{dt}] \quad (12)$$

unde: K_P este factorul de proporționalitate.

T_d este constanta de timp de derivare care reprezintă intervalul de timp prin care componenta derivativă anticipează efectul acțiunii proporționale. Constanta de timp de derivare prezintă avantajul comportării anticipative, dar are dezavantajul că amplifică semnalele de tip zgomot, putând cauza efecte de saturație în comportarea elementului de execuție.

Componenta derivativă arată faptul că mărimea de comandă $u(t)$ este proporțională cu viteza de variație a erorii. Această componentă derivativă nu poate fi utilizată niciodată singură, aceasta fiind efectivă numai în perioada tranzitorie (pentru $\varepsilon(t) = \text{constantă}$, comanda derivativă = 0).

În relația (13) funcția de transfer a controllerului proporțional derivativ trebuie explicitat și filtrul trece jos fiind obligatorie folosirea a cel puțin unui filtru pe intrarea controllerului derivativ prezent în [1].

$$H_R(s) = \frac{U(s)}{\mathcal{E}(s)} = K_P * (1 + sT_d) \frac{1}{1+sT_f} \quad (13)$$

unde T_f este constanta de timp a filtrului trece jos.

Valoare constantei de timp al filtrului se alege ca fracțiune din valoarea constantei de timp de derivare, cum se poate observa în ecuația (14) prezent în [1].

$$T_f = \frac{T_d}{N} \quad , \text{unde } N = 8 \div 20 \quad (14)$$

Cu această precizare relația (13) devine relația (15).

$$H_R(s) = \frac{U(s)}{\mathcal{E}(s)} = K'_P \frac{1+sT_d}{1+s\frac{T_d}{N}} \quad (15)$$

De menționat este faptul că algoritmul de reglare PD, așa cum evidentiaza relația (12), reprezintă un sistem nerealizabil fizic. Cel mai adesea, algoritmul de reglare PD este definit prin funcția de transfer (15) prezent în [1].

O altă exprimare a algoritmului PD este relația (16) prezent în [1].

$$H_R(s) = \frac{U(s)}{\mathcal{E}(s)} = K_P \left(1 + \frac{sT_d}{1 + s\frac{T_d}{N}} \right) \quad , \text{unde } N = 10 \div 20 \quad (16)$$

Răspunsul indicial la un controller PD cu filtru de intrare este dat în Figura 7 prezent în [1].

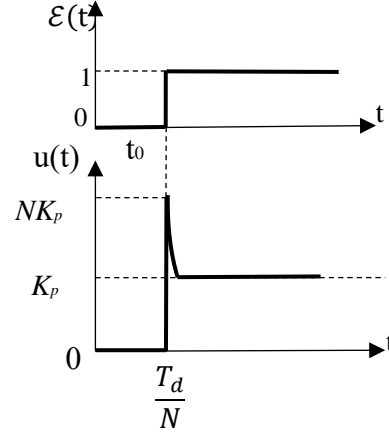


Figura 7. Răspunsul indicial al controllerului proporțional-derivativ prevăzut cu filtru la intrare

Efectul componente derivate este energic în primele momente ale răspunsului fiind sub forma unui salt ce se apropie de un impuls Dirac „derivative kick”. Limitarea acestui salt la valoarea $K_P \frac{T_d}{T_f}$ este efectul benefic al filtrului. Componenta derivativă este continuată de componenta proporțională având valoarea constantă egală cu K_p .

Impulsul inițial, datorită componente derivate, poate duce la saturarea ieșirii controllerului, astfel apare un efect de modulare în durată a impulsului inițial funcție de amplitudinea trepte de eroare aplicată la intrare. Acest efect este prezentat în Figura 8 prezent în [1].

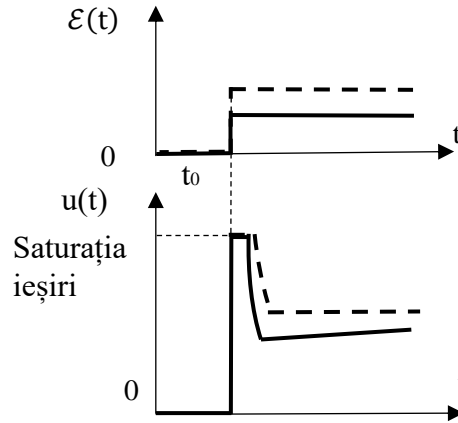


Figura 8. Efectului de modulare în durată a impulsului inițial datorat saturării ieșirii controllerului PD

2.1.4 Algoritm de reglare tip PID (proporțional-integral-derivativ)

La componeta de tip proporțională (P) dacă se adaugă încă două componente de tip integral (I) și de tip derivativ (D), rezultă un algoritm proporțional-integral-derivativ a cărui acțiune caracterizează mărimea de comandă $u(t)$ ca o suprapunere a trei componente, respectiv proporțională cu eroarea $\varepsilon(t)$, proporțională cu integrala erorii și proporțională cu derivata erorii, exprimându-se prin relația (17) prezent în [1].

$$u(t) = K_p[\varepsilon(t) + \frac{1}{T_i} \int_0^t \varepsilon(\tau) d\tau + T_d \frac{d\varepsilon(t)}{dt}] \quad (17)$$

unde: K_p este factorul de proporționalitate, T_d este constanta de timp de derivare, T_i este constanta de timp de integrare.

Relația (17) definește un algoritm PID ideal și se exprimă prin funcția de transfer (18) prezent în [1].

$$H_R(s) = \frac{U(s)}{\varepsilon(s)} = K_p * (1 + \frac{1}{sT_i} + sT_d) \quad (18)$$

Și în cazul acestui controller se impune prezența a cel puțin a unui filtru trece jos pentru filtrarea componentei derivative. Exprimarea funcției de transfer a controllerului PID cu filtru trece jos va avea forma (19) prezent în [1].

$$H_R(s) = \frac{U(s)}{\varepsilon(s)} = K_p * (1 + \frac{1}{sT_i} + \frac{sT_d}{1+s\frac{T_d}{N}}) \quad ,unde: N = 10 \div 20 \quad (19)$$

Prin introducerea filtrului trece jos pe canalul erorii acesta va influența toate componentele algoritmului funcția de transfer modificată a controlărilor PID va fii relația (20) prezent în [1].

$$H_R(s) = \frac{U(s)}{\varepsilon(s)} = K_p * \left(1 + \frac{1}{sT_i} + sT_d\right) \frac{1}{1+s\frac{T_d}{N}} \quad ,unde: N = 5 \div 20 \quad (20)$$

Algoritmul PID exprimat prin relațiile (17),(18) și (19) poartă numele de algoritm PID „fără factor de interacțiune” deoarece între anumite limite parametrii de acord ai controllerului, respectiv K_p , T_i și T_d pot varia independent fiecare cea ce corespunde unei implementări de tip paralel a componentelor algoritmului P,I și D.

În locul parametrilor T_i și T_d pot fii utilizați factorii de aplicare: $K_i = \frac{K_p}{T_i}$ pentru acțiunea componentei integrale și $K_d = K_p * T_p$ pentru acțiunea componentei derivative.

Astel algoritmul PID fără factor de interacțiune se exprimă în timp prin formula (21) prezent în [1].

$$u(t) = K_p \varepsilon(t) + K_i \int_0^t \varepsilon(\tau) d\tau + K_d \frac{d\varepsilon(t)}{dt} \quad (21)$$

În variabila Laplace exprimarea funcției de transfer va fii ecuația (22) prezent în [1].

$$H_R(s) = \frac{U(s)}{\varepsilon(s)} = K_p + \frac{K_i}{s} + \frac{sK_d}{1+s\frac{T_d}{N}} \quad unde N = 8 \div 20 \quad (22)$$

O structură de tip paralel a acontrollerului fără factor de interacțiune se poate vede în Figura 9 prezent în [1].

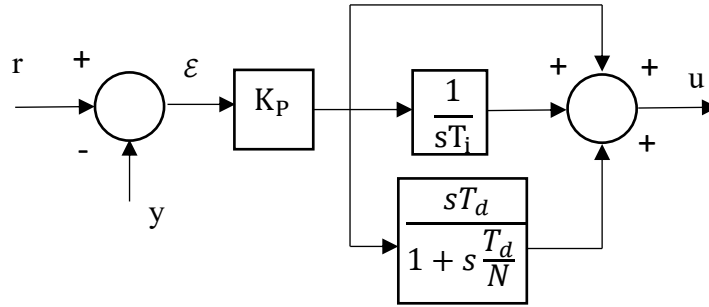


Figura 9. Structura de tip paralel a controllerului fără factor de interacțiune

O altă variantă puțin diferită, față de un algoritm PID ideal, este algoritmul PID cu factor de interacțiune care se exprimă prin funcția de transfer în relația (23) prezent în [1].

$$H_R^*(s) = K_p^* \left(1 + \frac{1}{sT_i^*} \right) (1 + sT_d^*) \quad (23)$$

Având în vedere efectul filtrului trece jos, imperios necesar limitării acțiunii componente derivative, relația care definește funcția de transfer a controllerului de tip PID cu factor de interacțiune va fii ecuația (24) prezent în [1].

$$H_R^*(s) = K_p^* \left(1 + \frac{1}{sT_i^*} \right) \frac{1 + sT_d^*}{1 + s \frac{T_d^*}{N}} \quad , \text{unde } N = 8 \div 20 \quad (24)$$

care corespunde implementări de tip serie structură obținută cu Figura 10 prezent în [1].

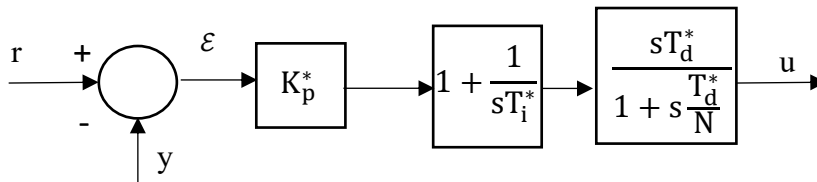


Figura 10. Structura de tip serie a controllerului fără factor de interacțiune

Dacă analizăm atent relațiile de reprezentare a funcțiilor de transfer ajunge la concluzia că prin aplicarea unor corespondențe între termeni relațiilor, respectiv între parametri de acord acelor două reprezentări, putem oricând reprezenta controllerul PID cu factor de intracțiune ca și un controller PID fără factor de interacțiune, substituind astfel cu relațiile (25) prezent în [1].

$$K_p = K_p^* \frac{T_i^* T_d^*}{T_i^*} \quad ; \quad T_i = T_i^* + T_d^* \quad ; \quad T_d = \frac{T_i^* T_d^*}{T_i^* + T_d^*} \quad (25)$$

Totodată procedeul poate fii aplicat și vice versa prin aplicarea următoarelor corespondențe între parametri (26) prezent în [1].

$$K_p^* = \frac{1}{2} K_p (1 + \sqrt{1 - \frac{4T_d}{T_i}}); \quad T_i^* = \frac{1}{2} T_p (1 + \sqrt{1 - \frac{4T_d}{T_i}}); \quad T_d^* = \frac{1}{2} T_d (1 + \sqrt{1 - \frac{4T_d}{T_i}}) \quad (26)$$

Din analiza relațiilor, observăm că se poate echivala controllerul PID fără factor de interacțiune cu controllerul PID cu factor de interacțiune dacă este îndeplinită obligatoriu condiția $T_i \geq 4T_d$.

De aici se vede clar, că un controller fără factor de interacțiune are o plajă mai mare de acoperire și e mai puternic în manifestările specifice controllerului PID expresia fiind mai reprezentativă, mai des folosită când se vorbește despre algoritmul PID.

Totuși controllerul PID cu factor de interacțiune nu este neglijat când ușurița la acordarea manuală. La exprimarea diferitelor variante ale controllerului PID respectiv la înlocuirea, alternarea tipului de controller, trebuie obligatoriu a fii reajustate valorile parametrilor de acord. Formele de tip cu sau fără factor de interacțiune vor fii diferite numai la utilizarea controllerului de tip PID, la ceilalți algoritmi de tip P, PI, PD, cele două forme cu sau fără interacțiune fiind aceleași.

Se poate observa răspunsul indicial al controllerului proporțional derivativ prevăzut cu filtru în Figura 11 prezent în [1].

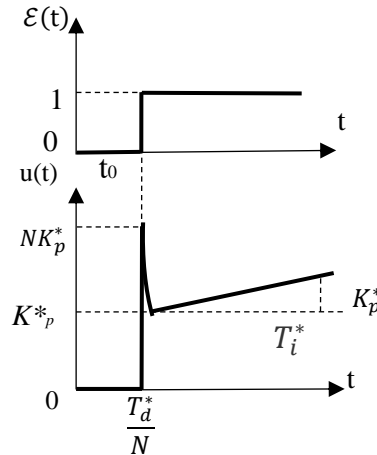


Figura 11. Răspunsul indicial al controllerului proporțional integral derivativ prevăzut cu filtru

Prin reprezentarea răspunsului indicial al controllerului PID prevăzut cu filtru, observăm că efectul derivativ se manifestă primul, apoi urmează efectul proporțional și la urmă efectul integral.

Componenta derivativă are un salt esențial reproducând cu aproximație impulsul Dirac. Componenta proporțională se impune ca un scurt palier la valoarea K_p^* , iar componenta integrală va fii ca o rampă crescătoare.

Tot atunci, impulsul inițial creat de componenta derivativă se poate limita prin saturarea ieșirii controllerului, impulsul inițial având durată proporțional mai mare în funcție de amplitudinea treptei aplicate pe canalul erorii. Efectul de modulare în durată a impulsului inițial este utilizat la forțarea părți inerțiale electromecanice a buclei de reglaj.

Totuși în cazul unei erori nenule persistente, se poate întâmpla ca componenta integrală să genereze fenomenul ne dorit de saturare a ieșirii elementului de execuție.

Se poate observa efectul de modulare în durată a impulsului inițial datorat saturării ieșirii controllerului PID în Figura 12 prezent în [1].

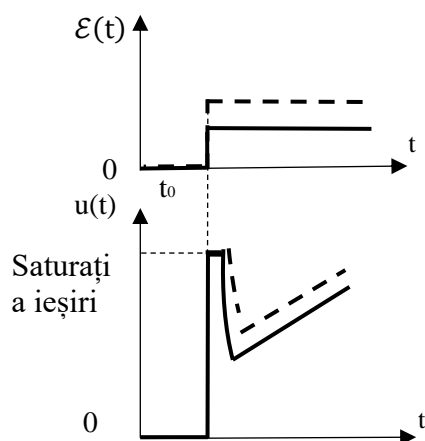


Figura 12. Efectul de modulare în durată a impulsului inițial datorat saturației ieșirii

2.2 Reglarea regimului termic la cuptoare

Temperatura interioară din cuptor se reglează manual sau automat. Prin reglarea automată a temperaturii la cuptoare electrice reduce consumul specific de energie electrică, reglajul asigură concordanța puterii absorbite de cuptorul electric cu regimul de temperatură. Sunt trei metode de reglare automată a temperaturii: continuă, semicontinuă și discontinuă. Cea mai complexă și scumpă este reglarea continuă, de aceea este utilizată mai rar. Cea mai simplă și mai ieftină dintre metode este reglarea discontinuă a temperaturii, asigură o precizie relativă de 1...3% , și practic se folosește sub forma reglării bipoziționale în majoritatea cazurilor. Reglarea bipozițională a temperaturii, ca metoda cea simplă și uzuală se realizează prin modificarea în două trepte a puterii cuptorului, făcând cuplarea și decuplarea alternativă a elementelor încălzitoare de la rețeaua electrică.

Variațiile temperaturii și puterii la reglarea bipozițională sunt prezentate în figura [13]

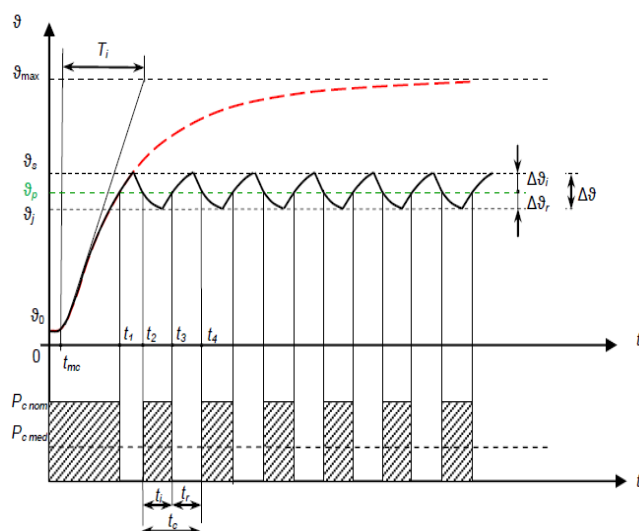


Figura 13. Variația temperaturii și a puterii la reglarea bipozițională

Sa considerat t_{mc} timpul mort al cuptorului, timpul în care nu se modifică temperatura deși se modifică curentul prin elementele încălzitoare. În lipsa reglajului, după $(4...5) T$, (T fiind o constantă

de timp pentru regimul de încălzire) temperatura va ajunge în regim staționar ϑ_{max} , evoluție prezentată în figura [13] prezent în [4].

Constanta de timp T se va determina grafic, fiind la intersecția dreptei de regim staționar ϑ_{max} și tangenta trasată la curba de încălzire. La reglarea bipozițională, dacă temperatura sesizată de traductor este mai mică decât temperatura prescrisă ϑ_p , regulatorul comandă alimentarea elementului încălzitor al cuptorului. Peste temperatura prescrisă la momentul t_1 regulatorul va comanda întreruperea alimentării din cauza inerției termice a cuptorului crește temperatura la ϑ_s urmând a scădea iar la momentul $t_2 < \vartheta_p$ prin comanda regulatorului se realimentează elementul încălzitor. Din nou inerția termică a cuptorului determină scăderea temperaturii la ϑ_j , urmând a crește din nou procesul reluându-se ciclic, temperatura interioară cuptorului variând între ϑ_j și ϑ_s . Totodată și puterea absorbită din rețeaua electrică va avea o evoluție ciclică cu durata t_c între valorile : puterea nominală (pe durata t_i) și zero (pe durata t_r), rezultând puterea medie în ecuația (27) prezent în [4].

$$P_{cmed} = \frac{P_{cnom} * t_i}{t_c} \quad (27)$$

În practică regulatorul bipozițional are o zonă e insensibilitate (regulator cu histerezis), iar traductorul de temperatură fiind o constantă de timp care depinde de construcția și felul materialelor componente, etc.

În fugura [14] se prezintă reglajul bipozițional cu variația temperaturii și a puteri cupturului la temperatura din incintă (linie continuă) și cea sesizată de traductorul de temperatură (linie întreruptă) prezent în [1].

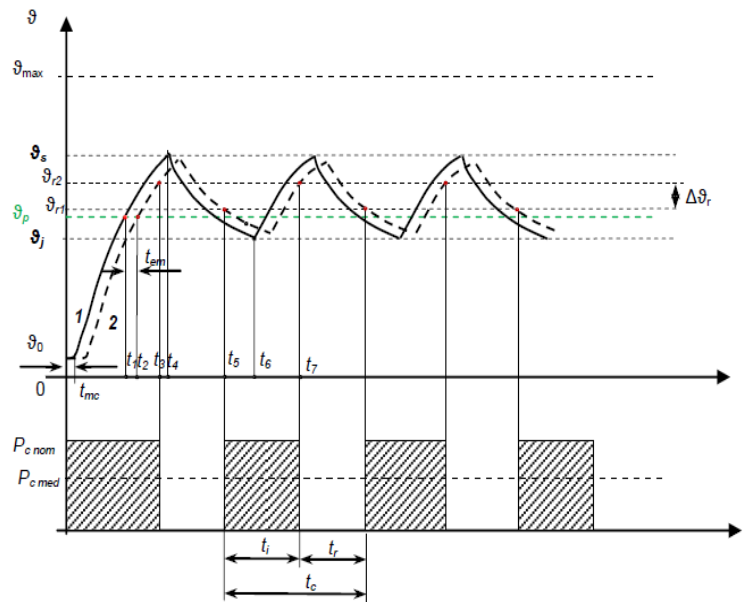


Figura 14. Variația temperaturii și a puterii, folosind regulator bipozițional cu histerezis și traductor cu constantă de timp nenulă

Regulatorul bipozițional cu histerezis indică fereastra de insesibilitate cu intervalul $\Delta\vartheta_r$, aproximativ temperatura prescrisă ϑ_p care se atinge la momentul t_1 dar traductorul sesizând-o doar la momentul t_2 datorită constatei proprii de timp. Regulatorul va acționa abia la mometul t_3 când se atinge temperatura ϑ_{p2} , comandă deconectare elementului încălzitor. Din cauza inerției termice temperatura în cuptor crește până la t_4 , urmând a descrește.

Variația temperaturii este sesizată cu întârziere de traductor, reconectarea elementului încălzitor făcându-se la momentul t_5 când se ajunge la temperatura ϑ_{p2} . Datorită inerției termice temperatura va scădea la momentul t_6 , după care se repetă. La momentul t_7 se comandă deconectarea.

2.2 Termocuplu tip K

Termocuplul de tip K este cel mai utilizat în industrie, este fabricat dintr-un amestec de fire de obicei nichel, crom sau aluminiu. Acesta este format dintr-un metal nemagnetic și un metal magnetic.

Caracteristică specifică este apariția unei abateri la ieșire când materialul magnetic atinge punctul de temperatură la care un material se demagnetizează.

Compoziția materialului a ramurii pozitive este formată din 90% nichel, 10% crom, iar ramura negativă este formată din 95% nichel, 2% aluminiu, 2% mangan și 1% siliciu.

Avantajele folosirii termocuplului de tip K este rezistența în condiții de funcționare în medii dure, răspuns rapid, durata de viață mai ridicată față de alte termocupluri și datorită compoziției de fire crome și alumele poate să ajungă la o variație de temperatură de la $-270\text{ }^{\circ}\text{C}$ la $1260\text{ }^{\circ}\text{C}$ și rezistența la oxidare.

Se poate vedea o schemă a termocuplului de tip K cu polaritățile și materialul din care este confecționat în Figura 15, prezent în [7].

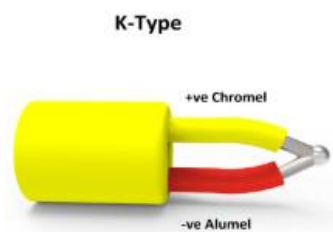


Figura 15. Termocuplu de tip K și polaritățile lui

3 Implementarea soluției adoptate

3.1 Ponderea mărimii prescrise la algoritmul PID

La controllerul PID fără factor de interacțiune la care mărimea de comandă $u(t)$, are șocuri (impulsuri nedorite) când prescrierea are o variație rapidă (în trepte). Șocurile se datorează acțiunii componente derivate asupra erorii. Deoarece eroarea este diferența dintre mărimea prescrisă și variabila de ieșire din proces $\mathcal{E}(t) = r(t) - y(t)$ prin derivarea erorii în relația (28) prezent în [1].

$$\frac{d\mathcal{E}(t)}{dt} = \frac{dr(t)}{dt} - \frac{dy(t)}{dt} \quad (28)$$

Dacă prescrierea are o variație, treaptă de semnal vom avea $\mathcal{E}(t) = \sigma(t)$ prin a cărei derivare obținem impulsul Dirac. Dar derivata ieșiri a procesului $\frac{dy(t)}{dt}$ nu va introduce impulsuri în mărimea de comandă, deoarece evoluția în sistemul inerțial, semnalul $y(t)$, va avea variabilă lentă de timp față de prescriere. Reducerea șocurilor datorate componente derivate a controllerului PID se realizează prin ponderea prescrieri care se exprimă în relația (29) prezent în [1].

$$u(t) = K_p \{ \beta r(t) - y(t) + \frac{1}{T_i} \int_0^1 \mathcal{E}(\tau) d\tau + T_d [\gamma \frac{dr(t)}{dt} - \frac{dy(t)}{dt}] \} \quad (29)$$

Unde avem ponderi diferite de eroare prelucrată prin coeficienții sub unitari β și γ componeta proporțională și componenta derivativă. Eroarea dată de componenta integrată nu va fi ponderată pentru a păstra performanțele componentei în regim staționar. Șocurile introdu-se în mărimea de comandă vor scădea proporțional cu coeficientul cu pondere derivativ γ . Totodată, prin scăderea valorii coeficientului componentei proporționale β se va atenua saltul apărut în prescriere.

Se poate vedea structura unui controller PID cu ponderea mărimii prescrise în Figura 16 prezent în [1].

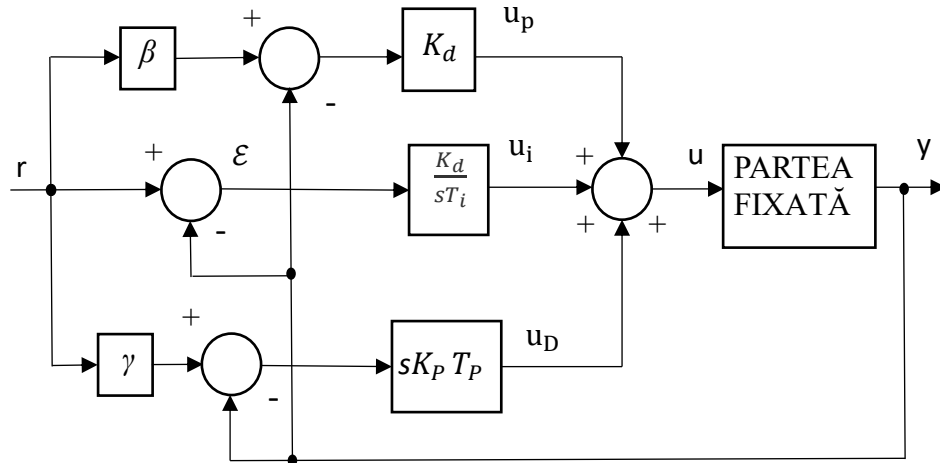


Figura 16. Structura unui controller PID cu ponderea mărimii prescrise

Dacă coeficienții $\beta = \gamma = 1$ vom avea algoritmul PID fără factor de interacțiune standard.

3.2 Utilizarea algoritmului PID

Cel mai utilizat algoritm de reglare din aplicațiile industriale este algoritmul PID apărut prin controllerul pneumatic în 1925. Algoritmul PID are diferite forme: de sine stătător sau ca parte a sistemelor DDC, sau sistemele complexe de conducere ierarhizate, unde algoritmul PID are nivele inferioare. Un algoritm PID este un dispozitiv cu regului empirice sau un element cu modelare analitică. Prin utilizarea controllerului PID se obține rezultate foarte bune în raport cu cantitatea redusă de informații disponibile la proiectarea bucle de reglare automată. Controllerele PID în timp au trecut prin multe schimbări tehnologice cum ar fi: implementări mecanice, hidraulice, implementări cu microprocesoare, tehnologii cu tuburi electronice cu vid, tehnologii cu tranzistoare și circuite integrate analogice. Față de alte timpuri controllerele bazate pe microprocesoare au avantaje suplimentare ca auto acordarea, programarea parametrilor, respectiv reactualizarea continuă a parametrilor. Controllerul PID se poate utiliza în sisteme cu o dinamică de ordinul I (cu un singur pol), performanțele fiind asigurate chiar și în lipsa componentei derivative. Pentru sisteme de ordinul II sau ordin mai mare, este utilizat controllerul PID, componenta derivativă având rolul de a reduce timpul de răspuns și de a menține factorul de proporționalitate utilizat sub o valoare maximală, cu care sistemul ar ajunge la limita stabilității.

La sistemele cu timp morți de valori reduse, se va îmbunătăți stabilitatea buclei de reglare prin utilizarea componentei derivative având efectul de anticipație.

3.3 Triacul (BTA16B)

Dispozitivul semiconductor de putere cu șase straturi cu comanda intermitentă, în curent, este un triac. Structura triacului se constituie din două bucle cu reacție pozitivă, caracteristică structurii în patru straturi dispozitivul având proprietatea de agățare.

În Figura 17 este ilustrat structura internă a triacului, schema echivalentă și simbolul triacului, prezent în [2].

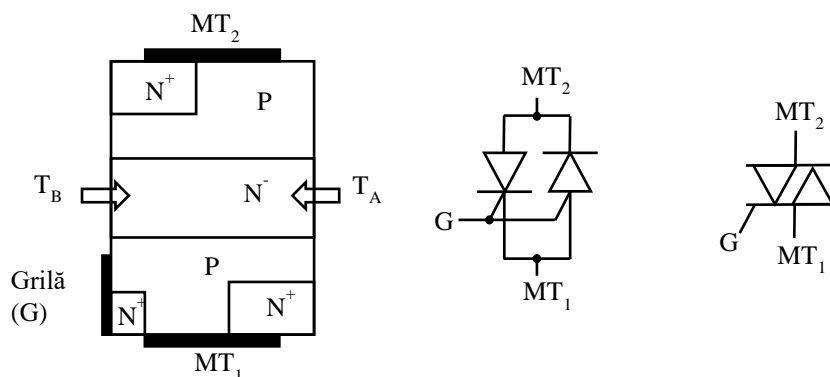


Figura 17. Structura internă, schema echivalentă și simbolul triacului

Simbolul și schema echivalentă arată că triacul are două tiristoare în conexiune antiparalelă respectiv cu grila catodică avem tiristorul T_A și cu grila anodică avem tiristorul T_B .

Cele două tiristoare, T_A și T_B au grile conectate împreună la terminalul de control al dispozitivului (grila G). În această configurație de conectare a celor două tiristoare este posibilă folosirea triacului în

controlul puterii care se transferă la sarcină în curent alternativ, pe amândouă alternanțele tensiunii alimentate.

Terminalele MT_1 și MT_2 de forță, ale triacului poartă denumirea de terminale principale și nu se mai numesc anod și catod.

Terminalul de referință pentru curenții și tensiunile triacului este terminalul MT_1 .

În Figura 18 este reprezentată caracteristica statică de ieșire a triacului. Terminalul MT_2 este pozitiv în raport cu MT_1 , în cadranul I, iar în cadranul III MT_2 este negativ în raport cu MT_1 prezent în [2].

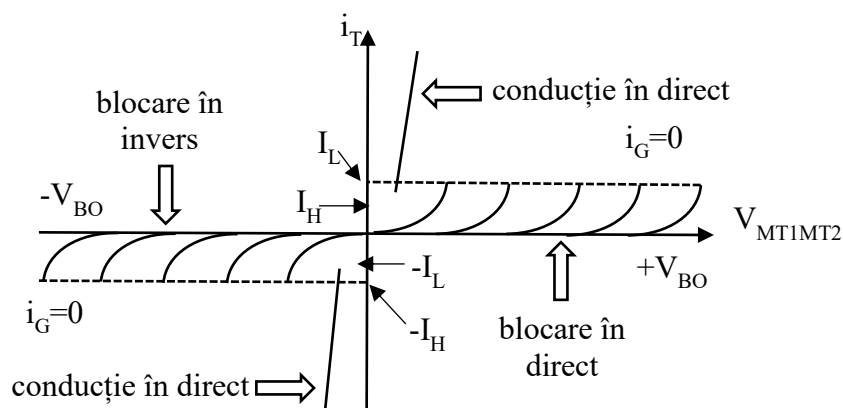


Figura 18. Caracteristica statică de ieșire a unui triac

Dacă se vrea controlul de amorsare a triacului prin curenți de grilă în oricare dintre cadrane, valoarea tensiunii de străpungere în direct V_{BO} (break over voltage) nu va fi depășită nici în regimul de polarizare inversă.

Dacă accidental valoarea tensiunii de străpungere V_{BO} este depășită atunci triacul intră în conducție și rămâne blocat păstrându-și starea până ce valoarea curentului prin terminalele principale va scădea sub valoarea curentului de menținere I_H .

Astfel triacul nu va avea nevoie de circuite auxiliare pentru protecție la supratensiune pentru că amorsează la fiecare creștere peste valoarea tensiunii de străpungere V_{BO} .

Totuși dacă acest fel de amorsare nu este dorită, fiind considerată o amorsare parazită, deși dispozitivul nu se deteriorează, triacului i se pot prevedea circuite de protecție pentru stoparea declanșărilor intempestive.

Caracteristica statică a triacului se poate echivala cu un comutator electronic bidirecțional, datorită simetriei sale, din acest motiv triacul are utilizarea în circuitele cu curent alternativ.

În cadranul I triacul funcționează pe alternanță pozitivă a tensiunii aplicate la terminalele principale iar în cadranul III vom avea funcționarea pe alternanță negativă.

În ambele cadrane caracteristica statică va avea o ramură corespunzătoare stării de blocare și altă ramură pentru starea de conducție.

Modificarea din blocaj în conducție respectiv amorsarea triacului se face prin comanda în grilă.

Caracteristici ale triacului BTA16B:

- curent de suport de capacitate mare
- comutare $(dV/dt)_c > 10\text{V}/\mu\text{s}$
- tensiunea de izolare 2500V

Dispozitivele de tip PNP pasivate din sticlă de înaltă performanță fac parte din familia triac BTA/BTB16 B. Acestea sunt utilizate pe scară largă în aplicațiile care necesită capacitate mare de

supratensiune ca de exemplu aplicații: pentru control de fază, comutate statică la o sarcină rezistivă sau inductivă. Pinii triacului se pot observa în Figura 19, prezent în [12].

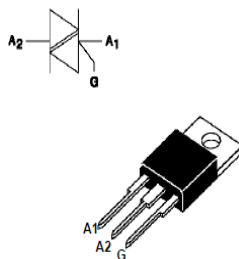


Figura 19. Triact BTA16B și simbolul lui

3.4 Termocuplu (MAX6675)

Max6675 face compensarea capătului rece și digitalizează semnalul de la termocuplul K. Datele citite au o rezoluție de 12 biți doar în format pentru citire, compatibil SPI. Convertorul digital are domeniul de temperatură între 0,25°C până la 1024°C, are precizia termocuplului de 8LSB pentru temperaturi de la 0°C și +700°C. Este disponibil în pachet mic cu 8 pini și este utilizat în aplicații industriale, aparate electrice și instalații climatizare.

Caracteristici:

- conversia digitală directă a termocuplului de tip K la ieșire
- compensarea joncțiunii reci
- interfață serială simplă compatibilă cu SPI
- rezoluție de 12 biți, 0,25°C

Evaluări maxime absolute:

- Tensiune de alimentare (VCC la GND): -0,3V la +6V
- Curent SO: 50mA
- Protecție ESD (Model de corp uman): ±2000V
- Disiparea continuă a puterii (TA = +70°C)
- SO cu 8 pini (reducere 5,88 mW/°C peste +70°C): 471 mW
- Interval de temperatură de funcționare: -20°C până la +85°C
- Interval de temperatură de depozitare: -65°C până la +150°C
- Temperatura joncțiunii: +150°C
- Faza de vapori (60s): +215°C
- Infrarosu (15s): +220°C
- Temperatura plumbului (lipire, 10s): +300°C

MAX6675 este un convertor analog digital de 12biți încorporat în ADC. MAX6675 este proiectat să funcționeze cu un microcontroller extern sau un alt termostat inteligent, în aplicații de control sau monitorizare.

Max6675 conține semnal condițional hardware pentru a convertii semnalul termocuplului într-o tensiune compatibilă cu canalele de intrare ale ADC-ului. Cele două intrări T+ și T- conectate prin circuitele interne, reduc erorile de zgomot de la firele termocuplului.

Înainte de convertirea tensiunii termoelectrice la valori echivalente de temperatură, va fi necesară compensarea diferenței dintre joncțiunea rece a termocuplului (temperatură ambiantă) și o valoare de referință de 0°C (temperatură virtuală). Tensiunea se va modifica la termocuplu tip K, cu 41μV/°C aproximându-se caracteristica termocuplului cu ecuație liniară (30), prezent în [8]

$$V_{OUT} = (41\mu V / ^\circ C) * (T_R - T_{AMB}) \quad (30)$$

Unde:

V_{OUT} este tensiunea de ieșire a termocuplului (μV).

T_R este temperatura joncțiunii termocuplului de la distanță (°C).

T_{AMB} este temperatura ambiantă (°C).

Rolul termocuplului este detectarea diferenței de temperatură între două capete. Capătul fierbinte va avea valori în intervalul de temperatură de la 0°C până la 1023,75°C. Capătul rece este la temperatura mediului ambiant (a plăcii de montaj) putând avea valori în intervalul de la -20°C până la 85°C.

Chiar dacă temperatura ambiantă are variații, MAX6675 arată cu precizie diferența de temperatură la capătul opus. MAX6675 va determina și corecta variația temperaturii ambiante cu compensarea joncțiunii reci.

El convertește citirea valorii temperaturii ambiante cu o diodă cu senzor de temperatură, într-o tensiune. Se va măsura tensiunea la ieșirea termocuplului și a diodei de detectare, indicând temperatura efectivă a termocuplului.

Prin circuitul intern se transmite tensiunea diodei, arătată de temperatura mediului ambiant și tensiunea termocuplului, arătată de diferența de temperatură între capete, la funcția de conversie stocată în ADC, putând calcula temperatura capătului fierbinte al termocuplului.

Performanța optimă este atinsă când capătul rece al termocuplului și MAX6675 vor avea aceeași temperatură. Se va evita amplasarea dispozitivelor generatoare de căldură în apropierea MAX6675, deoarece se pot produce erori datorate joncțiunii reci.

ADC adaugă valorile citite cu dioda în joncțiunea rece cu tensiunea amplificată la termocuplu, citind rezultatul de 12 biți pe pinul SO. O succesiune de zerouri arată o temperatură de 0°C la termocuplu. O succesiune de numere 1 arată o temperatură de +1023.75°C la termocuplu.

MAX6675 procesează citirea de la termocuplu și transmite aceste date printr-o interfață serială, el având ca interfață un microcontroller.

Dacă forțăm CS scăzut și aplicăm semnalul de ceas la SCK, vom putea citi rezultate la SO. Forțarea CS scăzut va opri procesul de conversie vom iniția un alt proces de conversie forțând CS ridicat. Prin forțarea CS scăzut vom scoate pe pinul SO primul bit. Citirea completă a interfeței seriale se realizează la 16 cicluri de ceas, se vor citi biții de ieșire pe marginea descendentă a ceasului. Primul bit (D15) va fi de semn fals fiind totdeauna zero. Biții de D14 – D3 indică temperatura convertită în ordine MSB la LSB. Bitul D2 va fi de obicei scăzut și va deveni ridicat dacă intrarea termocuplului este deschisă. D1 este scăzut pentru a furniza ID-ul dispozitivului MAX6675 și bitul D0 având trei stări.

Bit-ul D2 este scăzut în mod normal și devine ridicat dacă este deschisă intrarea termocuplului. În ordine să permită funcționarea, detectării deschiderii termocuplului T trebuie neapărat înpământat.

Precizia MAX6675 este îndeolnică datorită zgomotului făcut de sursa de alimentare. Efectul zgomotos a sursei de alimentare poate fi atenuat dacă se folosește un condensator ceramic având 0.1μF ca bypas în apropierea de alimentare a dispozitivului.

Încălzire automată degradează precizia măsurării temperaturii a MAX6675 în unele aplicații. Mărimea erorii de temperatură este dependentă: de conductivitatea termică a MAX6675, de precizia de asamblare a dispozitivului (tehnica de montaj) și de efectul fluxului de aer. Pentru obținerea calității și a preciziei se recomandă folosirea unei plăci mai mari de montaj.

Totodată precizia sistemului de termocuplu se va îmbunătăți prin respectarea măsurilor de siguranță:

- Se recomandă folosirea firelor mai lungi care nu deviază căldura din zona măsurată
- Dacă este nevoie de un fir scurt, va fi utilizat stric în zona de măsurare iar pentru zona fără gradient de temperatură vom folosi un cablu prelungitor.
- Trebuie evitate frecările și vibrațiile pentru a nu tensiona firele.
- La utilizarea unui cablu lung la termocuplu, acesta va fi de tip multifilar.
- Evitați variațiile bruște de temperatură.
- Legăturile termocuplului vor fi utilizate numai sub temperaturile maxime proprii de funcționare.
- Firul termocuplului va fi adecvat mediului de utilizare.
- Cablul prelungitor se va utiliza numai la temperaturi scăzute și variații mici de temperatură.
- Se va înregistra evoluția termocuplului într-un jurnal.

Amplificatorul (A1) de intrare este cu zgomot redus, fiind conceput pentru a detecta intrările de precizie înaltă. Termocuplu și firele de conectare vor fi ținute la distanță față de sursele de zgomot.

Se poate observa diagrama bloc a MAX6675 în Figura 20, prezent în [8]

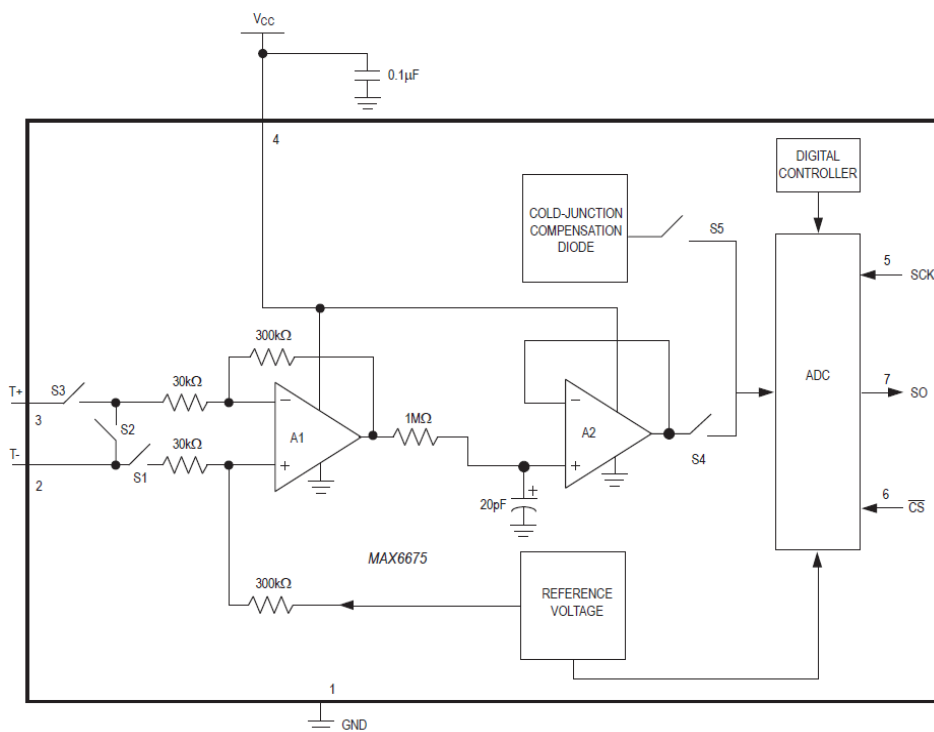


Figura 20. Diagrama bloc pentru MAX6675

3.5 Arduino Nano

Arduino Nano este placă de microcontroller având sursa deschisă, compatibilă cu placa, bazat pe microcontrollerul microchip ATmega328P (MCU) și dezvoltat de Arduino.cc lansat inițial în anul 2008.

Arduino Nano oferă conectivitatea și specificațiile plăci arduini uno într-o formă mai mică. Acesta are 30 anteturi I/O masculine, cu configurație asemănătoare DIP-30, care se pot programa

prin utilizarea mediului de dezvoltare integrat (IDE) Arduino Software, comun tuturor plăcilor arduino, rulând online și offline. Placa se alimentează prin cablu mini USB de tip B sau prin baterie de 9V.

Specificații tehnice:

- Microcontroller : Microcip ATmega328P
- Tensiune de funcționare: 5 volți
- Tensiune de intrare: 5 până la 20 volți
- Pini I/O digitale: 14 (6 ieșiri PWM opționale)
- Pini de intrare analogică: 8
- DC pentru pin de 3,3 V: 50 mA
- Memorie flash : 32 KB, din care 2 KB sunt folosite de bootloader
- SRAM : 2 KB
- EEPROM : 1 KB
- Viteza de ceas: 16 MHz
- Lungime: 45 mm
- Latime: 18 mm
- USB: Mini-USB de tip B

Arduno Nano asigură facilități de comunicare cu computer, un alt Arduino sau cu alte micro controllere. Comunicație serială UART TTL (5V) este oferită de ATmega328, care este disponibilă pe pinii digitali 0 (RX) și 1 (TX). De pe placă FTDI FT232RL canalizează comunicarea în serie prin USB, iar driverele FTDI incluse cu firmware-ul Arduino oferă portul comunicației virtuale pentru software-ul computerului. Arduino are un monitor serial prin care trimite datele textuale simple spre și de la placa arduino RX și TX. Led-urile de pe placă vor clipi atunci când se face transmiterea datelor prin cipul FTDI și conexiunea USB la computer, dar nu și atunci când avem comunicare în serie pe pinii 0 și 1. O bibliotecă poate permite comunicarea în serie pe oricare pin digital. Totodată ATmega328 acceptă comunicațiile I2C și SPI.

În Figura 21 este ilustrat o schema a plăcuței Arduino Nano și pinii lui, prezent în [10].

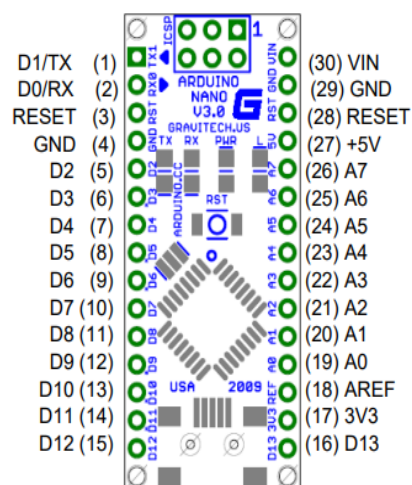


Figura 21. Pinii plăcuței Arduino nano

3.6 Dispozitiv driver triac izolat optic (MOC 3020)

Această componentă face parte din familia de drivere optoizolate pentru triacuri cu trecere prin zero reprezintă o soluție practică, accesibilă și eficientă pentru a facilita interfața între circuitele de control cu curent redus, cum ar fi porțile logice și microprocesoarele, și sarcinile de putere cu curent alternativ. Aceste componente furnizează o cantitate suficientă de curent pentru a declanșa tiristoarele cu curent și tensiune mari, în timp ce mențin o izolație dielectrică de încredere, cu o tensiune de 7.5 kV între linia de alimentare și circuitul de comandă.

În cadrul unui comutator integrat ce operează cu trecere prin zero și este localizat pe cipul detector, se realizează eliminarea fluctuațiilor de curent și a interferenței electromagnetice rezultate (EMI), precum și soluționarea problemelor de fiabilitate într-o gamă variată de aplicații. Cu o imunitate puternică la tranzitorii, măsurată la 5000 V/ μ s, împreună cu caracteristici ce minimizează cuplarea, o rezistență izolantă sporită și valori VDRM de până la 800 de volți, această serie de drivere pentru triacuri reprezintă o conexiune crucială între circuitele de control sensibile și mediul de furnizare cu curent alternativ.

Un aspect esențial de subliniat este că aceste drivere izolate optic pentru triacuri nu sunt proiectate pentru a funcționa de sine stătător, în mod similar cu dispozitivele cum ar fi releele cu stare solidă. În schimb, ele pot eficient înlocui circuitele de comandă discrete, costisitoare și voluminoase, care implică o serie de componente, cum ar fi optoizolatoare tranzistor standard, elemente de suport, inclusiv un pod redresor cu val de vârf complet, tranzistor individual, SCR-uri de declanșare și diverse combinații de rezistoare și condensatoare.

În ceea ce privește structura, aceste dispozitive se fundamentează pe utilizarea unui LED emițător de lumină infraroșie cu fază lichidă EPI, care activează optic un cip detector de siliciu. Ilustrația schematică a driverului pentru triacuri este prezentată în Figura 22 și prezent în [17]. Ambele cipuri sunt încorporate într-un pachet compact cu două rânduri de pini (DIP) și 6 pini, asigurând astfel stabilitate mecanică și protecție împotriva elementelor străine pentru cipurile semiconductoare. Pentru izolarea cipurilor, se utilizează un mediu transparent în infraroșu, care eficient separă circuitele de control ale LED-ului de sursa de alimentare cu curent alternativ. Acest sistem de izolare respectă rigorile cerințelor de izolație impuse de agenții de reglementare precum UL și VDE.

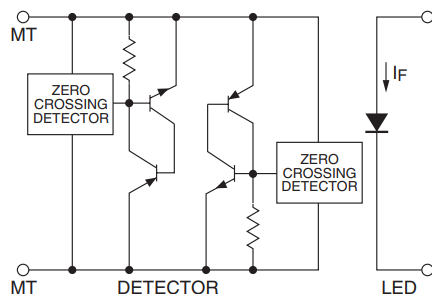


Figura 22. Schema unui driver Triac izolat optic cu trecerea la zero

În același timp, cipul detector reprezintă un circuit integrat monolitic complex, conținând două SCR-uri sensibile la infraroșu, cu tensiune înaltă, montate în paralel în direcții opuse, având rolul de a acționa ca un triac sensibil la lumină. Găurile individuale de declanșare ale acestor SCR-uri sunt conectate la circuite rapide de detectare a trecerii prin zero. Această configurație asigură că atunci când curentul continuu trece prin LED, detectorul nu va trece în starea de conducție până când tensiunea alternativă aplicată ajunge aproape de zero. Această particularitate nu numai că minimizează generarea de zgomot electromagnetic (EMI) și curentul de pornire (surge) în cazul sarcinilor rezistive și sarcinilor

cu inductanță moderată, dar furnizează și o imunitate înaltă la zgomot (de ordinul miilor de V/ μ s) pentru circuitul de detecție.

Diodele care emit infraroșu, cu arseniură de galiu cuplate optic la un comutator din silicon, bilateral, alcătuiesc seria MOC3020. Aceste dispozitive sunt testate și marcate corespunzător cerințelor VDE0884, deosebindu-se în cazul unor comenzi prin sufixul (V) care este obligatoriu inclus în terminația numărului piesei. VDE 0884 este o variantă de testare. Este proiectat pentru aplicații la care este necesară declanșarea izolată a triacului.

În Figura 23 este ilustrat conexiunile lui MOC 3020 și schematicul acestuia, prezent în [13].

Este recomandat la aplicații de 115/240Vac: ca alimentare cu comutator static, la balasturi lămpi, la releie cu stare solidă, la microprocesoare cu interfața periferică de 115Vac, comenzile motorului.

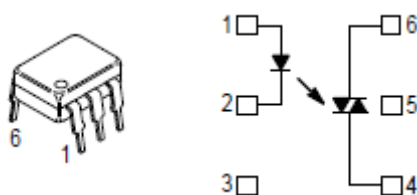


Figura 23. Schematicul componentei MOC 3020 cu conexiunile: 1.Anod; 2.Catod; 3. NC (no internal connexion); 4. Terminalul principal; 5.Subtrat; 6.Terminal principal

3.7 Optocupler (PC817)

PC817 reprezintă un optocupler ce conține atât un fototranzistor, cât și o diodă IR. În diverse configurații de circuite, filtrele au rol esențial în eliminarea zgomotului. Atunci când un circuit integrează o rezistență și un condensator, acest lucru conduce în mod constant la eliminarea zgomotului din semnalul de intrare. Cu toate acestea, valoarea rezistenței și a condensatorului este frecvent determinată de caracteristicile semnalului intern.

În cadrul circuitului optocuplerului PC817, semnalul zgomotos este captat de dioda IR de pe o parte și este transmis către cealaltă parte prin intermediul semnalului IR. Acest proces permite funcționarea circuitului conform designului său inițial. PC817 IC conține un LED și un fototranzistor care sunt interconectate optic. Astfel, semnalul poate fi transmis optic între partea de intrare și cea de ieșire fără a fi necesară o conexiune fizică.

Așadar, acest circuit integrat contribuie la izolarea circuitelor. De asemenea, este folosit pentru a suprima zgomotul din semnal, separând circuitele de curent continuu și cele cu tensiuni scăzute de circuitele cu curent alternativ și cele cu tensiuni ridicate. De asemenea, acest circuit integrat este utilizat pentru a gestiona tensiunile alternative sau cele de mare amplitudine printr-un semnal analogic sau digital de mică magnitudine.

Operația optocuplerului PC817 este simplă, cu toate acestea, modul de utilizare variază în funcție de specificațiile dispozitivelor utilizate. Pe partea de intrare, optocuplerul necesită un rezistor de limitare a curentului, în timp ce pe partea de ieșire, trebuie să conectăm pinul logic de ieșire la pinul de alimentare. După generarea semnalului infraroșu, starea logică se schimbă de la nivelul "unu" la nivelul "zero" din cauza modificărilor în fluxul de curent.

Dispozitivul se caracterizează prin:

- raportul curentului de transfer.
- tensiunea mare izolată între intrare și ieșire.
- pachetul compact al dispozitivului dual-in-line.
- îndeplinește cerințele conform UL fiind catalogat în dosarul E64380.

Dispozitivul se regăsește în aplicații:

- la terminalele calculatoarelor
- la aparate de sistem sau la instrumente de măsurare.
- la copiatoare, registre, aparate electrocasnice sau ventiloconvectoare.
- la aparataje având transmisi de semnale cu impedanța și potențialul diferit între circuite.

În Figura 24 este ilustrat schematicul componentei PC817 împreună cu numerotarea pinilor, prezent în [15].

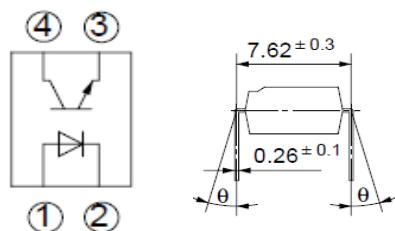


Figura 24. Schematicul componentei PC817 cu conexiunile 1.Anod; 2.Catod; 3.Emitor; 4.Colector

3.8 Modulul LCD 1602 având interfața serială I2C

Modulul de afișare LCD 16x2 este un modul de o înaltă calitate cu interfața de comunicare I2C, având două linii și 16 caractere, asigurând iluminarea de fundal și reglarea integrală a contrastului. Acest modul serial LCD I2C are avantajul real de simplificare a conexiunilor circuitului prin nefolosirea unor pini I/O pe placa arduino, simplifică dezvoltarea firmware cu biblioteca arduino care este disponibilă.

Modulul LCD 16x2 cu interfața I2C este compatibil cu placa arduino ca și cu alte plăci de control, având comunicarea I2C.

Date tehnice: -tensiunea de alimentare 5V.

- tipul de afișare este alb negativ pe iluminarea de fundal albastru.
- adresa I2C 0x38-0x3F.
- interfața I2C la 3 biți LCD date și linii de control.
- ajustarea contrastului cu potențiometrul încorporat.
- controlul lumini de fundal cu cablul jumper.
- dimensiunea plăci este 80x36mm.

LCD-ul disponibil pe scară largă și ieftin, cu un preț foarte avantajos, este cel cu nomenclatura și caracteristicile de tip HD44780 de la Hitachi. El are un rol deosebit în oricare din aplicații-le care afișează informații. Prin magistrala de comunicare I2C datele se pot afișa pe LCD, folosind placa piggy-back. Astfel de pachete există construite în jurul PCF8574, care este expander de port I/O bidirecțional de 8 biți de uz general, care folosește protocolul I2C. PCF8574 este circuitul CMOS din siliciu, oferind extindere I/O la distanța de uz general pentru majoritatea tipurilor de microcontrollere cu ajutorul magistralei bidirecționale cu două linii. Majoritatea modulelor piggy-back sunt bazate pe PCF8574T cu conexiunile de adresă A0-A1-A2, nu au conexiuni prin lipiere o să aibe adresa slave 0x27.

Adresa slave implicită 0x27 se va schimba în 0x3F, dacă pe placa de tip piggy-back are inclus un chip PCF8574AT.

În Figura 25 se poate vedea LCD-ul împreună cu modulul I2C, prezent în [11].



Figura 25. Ecranul LCD conectat cu modulul

3.9 Redresor (KBPC1010)

Puntea de diode este circuitul redresor cu 4 diode folosit în procesul de conversie a curentului alternativ în curent continuu, adică o polaritate fixă, de la bornele de intrare pe bornele de ieșire. Funcția punți de diode este convertirea tensiunii negative, a părți forme de undă în tensiune pozitivă. Un filtru trece-jos se poate utiliza pentru netezirea rezultatului în curent continuu. Cea mai utilizată aplicație pentru a face conversia intrării de curent alternativ în ieșire de curent continuu este redresorul în punte. Redresorul în punte realizează o redresare cu undă completă începând cu intrarea în curent alternativ având două fire. Această soluție are un cost și o greutate mai mică față de redresorul cu intrare cu 3 fire de la transformatorul având înfășurarea secundară centrală.

Înainte de apariția circuitului integrat, redresorul în punte sa constituit din diode separate. Astfel, cam din 1950 configurația de punte, având cele 4 diode conectate, ca o componentă cu 4 termin ale a fost utilizată cu succes, fiind disponibilă, având valori nominale de curent și tensiune.

Totodată la diverse aplicații diodele pot fii folosite în configurații de punte alături de condensatoare funcționând ca multiplicatori de tensiune.

Caracteristici : - Ideal for printed circuit boards;

- Capacitate mare de supratensiune directă;
- Mărime mică, greutatea mică de 5,62g și instalare simplă;
- Are polaritatea marcată pe corp;
- Curent de supratensiune de vârf de 250A;
- Tensiune de izolare de 2500V.

În Figura 27 este ilustrat schematicul piesei KBPC1010, prezent în [16], iar în Figura 26 este diagramă Gretz cu patru diode, prezent în [19].

Semiundă monofazată 60 Hz, sarcină rezistivă sau inductivă, pentru reducerea curentului de sarcină capacitiv cu 20%. Evaluări la temperatura ambiantă de 25°C.

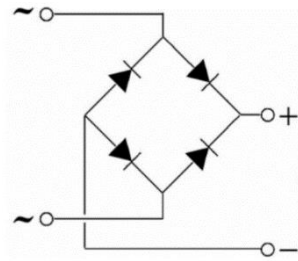


Figura 26. Schema KBC1010 văzută de sus din lateral

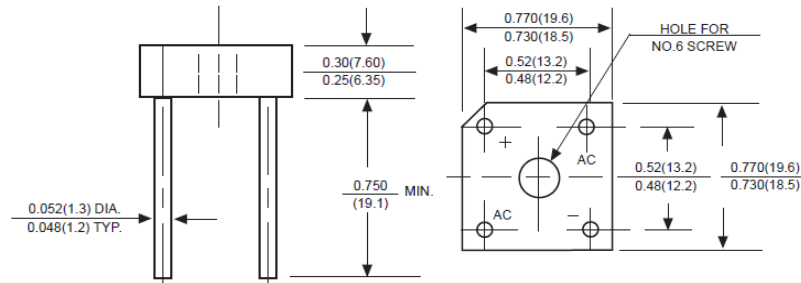


Figura 27. Schema KBC1010 văzută de sus din lateral

3.10 Codul Arduino folosit și explicat

Acest cod este prezent în anexă și comentat, dar am ales să îi fac o explicație mai detaliată.

Codul este pentru programul Arduino care controlează o buclă de reglare PID pentru a menține temperatura într-o anumită valoare (setpoint). Sistemul utilizează un termocul pentru a citi temperatura și este afișat pe LCD pentru a arăta valorile temperaturii și setpoint-ul. De asemenea se utilizează butoane pentru a mări sau scădea setpoint-ul.

La începutul codului sunt bibliotecile folosite aceste sunt pentru funcționalitatea termocuplului MAX6675, comunicarea I2C cu LCD și alte funcționalități Arduino.

Următoare parte definesc pinii Arduino la care sunt conectați pinii pentru declanșarea triacului, pinul pentru crucea zero, pinii pentru butoanele de creștere și scădere a setpoint-ului, și pinii pentru comunicația cu termocuplul.

Am folosit comanda standart din biblioteca *max6675.h*, această linie inițializează comunicarea cu termocuplul MAX6675 utilizând pinii specificați anterior.

Apoi am folosit comanda standart din biblioteca *LiquidCrystal_I2C.h* pentru a putea configura LCD-ul cu adresa specifică 0x27, acest lucru va porni LCD-ul.

Se definesc variabile pentru detectarea trecerii prin zero (*zero_cross_detected*) și pentru controlul încălzirii: un interval de întrerupere inițial (*firing_delay*) și un interval maxim de întrerupere (*maximum_firing_delay*).

Se mai definesc variabile pentru timp, aceste variabile stochează informații despre timpul trecut și curent, durata dintre citirile temperaturii (*temp_read_Delay*), temperatura reală citită de la termocuplu (*real_temperature*) și valoarea setpoint-ului dorit.

Se definesc și variabilele care sunt utilizate în calculul controlului PID, cum ar fi eroarea (*PID_error*), eroarea anterioară (*previous_error*), timpul dintre iterații (*elapsedTime*, *Time*, *timePrev*) și valoarea finală a controlului PID (*PID_value*).

Se definesc și constantele PID (P, I și D) care pot fi ajustate pentru a obține un control mai bun al temperaturii. Aceste constante se modifică în funcție de nevoile controllerului, pentru ele am inițializat și niște variabile care stochează componentele P, I și D ale controlului PID.

Am definit un buffer și un index pentru stocarea citirilor temperaturii pentru calculul mediei mobile.

Void setup configurează pinii folosiți în program „firing_pin” este utilizat pentru declanșarea triacului pentru a controla puterea căldurii, „zero_cross” acesta este folosit pentru a detecta momentul trecerii prin zero, când tensiunea alternativă trece prin zero volți „increase_pin” și „decrease_pin” sunt utilizați pentru a conecta butoanele de creștere și scădere a setpoint-ului.

Variabila „currentMillis” folosind funcția „millis()”, care returnează numărul de milisecunde de la pornirea plăcii Arduino.

Am adăugat un „if” care verifică dacă a trecut timpul necesar pentru a face o nouă citire a temperaturii de la termocuplu. Dacă timpul scurs de la ultima citire este mai mare sau egal cu temp_read_Delay, înseamnă că trebuie să facem o nouă citire a temperaturii. De asemenea, actualizăm previousMillis pentru a pregăti următoarea citire.

Următoare parte citește temperatura curentă de la termocuplu, valoarea citită este stocată în buffer și este incrementată pentru a trece la următoarea poziție în buffer și ne asigurăm că rămânem în limitele dimensiunii bufferului (5).

Apoi se calculează „PID_error = setpoint - real_temperature” care este eroarea PID ca diferența dintre setpoint (setpoint) și temperatura reală (real_temperature).

Următoarea secțiune verifică dacă eroarea PID este mai mare de 20. Dacă da, setăm componenta integrală (PID_i) la zero pentru a evita acumularea erorilor în exces în componenta integrală a controlului PID.

Urmează formula de calcul „PID_p = kp * PID_error” pentru componenta proporțională (PID_p) a controlului PID prin înmulțirea erorii PID cu constanta P (k_p).

După urmează formula de calcul „PID_i = PID_i + (ki * PID_error)” pentru componenta integrală (PID_i) a controlului PID prin adăugarea produsului dintre eroarea PID și constanta I (k_i) la valoarea anterioară a componente integrale.

Se măsoară timpul scurs de la ultima iterație și se calculează schimbarea erorii PID în funcție de timp. Acest lucru ajută la estimarea vitezei de variație a temperaturii și ajută la controlul temperaturii.

Formula de calcul „PID_d = kd * ((PID_error - previous_error) / elapsedTime)” pentru componenta diferențială (PID_d) a controlului PID.

Formula de calcul pentru valoarea finală a controlului PID (PID_value) adunând componentele P, I și D.

Urmează secțiunea care asigură că PID_value se încadrează în intervalul permis (între 0 și 7400). Această valoare trebuie să fie la fel ca și valoarea maximă pentru timpul de întârziere.

Urmează secțiunea unde se curăță ecranul LCD și afișăm informații relevante despre setpoint și temperatura reală măsurată. De asemenea, actualizăm previous_error pentru a fi utilizată în următoarea iterație a controlului PID.

Am implementat o secțiune verifică starea butonului de creștere a setpoint-ului (increase_pin). Dacă butonul este apăsat (LOW) și starea anterioară a butonului (button_increase_state) este falsă, se mărește valoarea setpoint-ului cu 5. Se actualizează starea anterioară a butonului pentru a evita modificarea setpoint-ului continuu atâta timp cât butonul este ținut apăsat.

Următoarea secțiune funcționează similar cu cea anterioară, dar verifică starea butonului de scădere a setpoint-ului (decrease_pin). Dacă butonul este apăsat (LOW) și starea anterioară a butonului (button_decrease_state) este falsă, se micșorează valoarea setpoint-ului cu 5. Se actualizează starea anterioară a butonului pentru a evita modificarea setpoint-ului continuu atâta timp cât butonul este ținut apăsat.

Se face verificarea dacă pinul `zero_cross` este la nivel înalt (HIGH), ceea ce indică detectarea trecerii prin zero. Dacă este adevărat, setăm variabila `zero_cross_detected` la `true`.

Apoi se face controlul de declanșare a triacului pentru a controla puterea căldurii. În funcție de valoarea PID calculată (`PID_value`), se așteaptă un anumit interval de timp pentru a declanșa triacul (`delayMicroseconds`), apoi se aplică o pulsare de 100 de microsecunde pe pinul „`firing_pin`” pentru a controla puterea. Variabila „`zero_cross_detected`” este setată la `false` pentru a pregăti următoarea trecere prin zero.

Funcția pentru calcularea mediei mobile cu numele „`calculateMovingAverage`” este o funcție separată pentru a calcula media mobilă a ultimelor 5 citiri ale temperaturii din bufferul `temperatureBuffer`. Funcția parcurge fiecare element din buffer, le adună și apoi împarte suma la 5 pentru a obține media.

4 Rezultate experimentale

În proiectul acest am folosit toate componente enumerate anterior. În aces proiect se fac următoarele : măsurare temperaturii cu un termocuplu de tip K, detectarea crucea zero a tensiunii AC, crearea controllerului PID și schimbarea unghiului de ardere la poarta triacului ,cu toate aceste elemente vom putea controla temperatura unui element de încălzire.

4.1 Rolul fiecărei componente

Pentru a putea controla partea pozitivă cât și partea negativă a semnalului AC este folosit un triac. Triacul va rămâne dezactivat până când primește un impuls la poarta sa și va rămâne activ până când intrarea principală își va schimba polaritatea.

Vom folosi triacul pentru a controla tensiunea AC, pentru a face acest lucru trebuie să detectăm crucea zero, deoarece pulsul nostru trebuie să fie în fază cu tensiunea AC și va trebui să detectăm când tensiunea trece de la pozitiv la negativ sau de la negativ la pozitiv și să fie sincronizat cu pulsul, astfel încât să se declanșeze mereu în același timp.

Pentru a putea detecta crucea zero folosim două componente acestea sunt redresor cu punte completă și un fotocupler. Lucrul acesta este realizat deoarece microcontrollerul nu poate funcționa cu tensiune negativă și cu tensiune mai mare de 5V. Redresorul va da doar undă pozitivă, iar fotocuplul va scădea tensiunea la $5V_{PP}$, făcând asta putem citi semnalul cu un Arduino. Am adăugat rezistența de $47k\Omega$ pentru a limita curentul și o rezistență de pull-down de $1k\Omega$.

Pentru a controla triac-ul este folosit un MOC3020 care este un dispozitiv de driver Triac izolat optic, conține o diodă emițătoare de infraroșu GaAs și un comutator bilateral de siliciu activat cu lumină, care funcționează ca un triac, datorită caracteristicilor sale de rezistență negative care îi permit să pornească rapid odată ce este atins un anumit nivel de tensiune aplicat.

Ecranul LCD este conectat cu un protocol de comunicare I2C. Conexiunea cu I2C are avantajul de a reduce numărul de pini necesari pentru controlul ecranului, deoarece folosește doar doi pini pentru a transmite datele și semnalele de control.

Rolul conectării prin protocolul I2C este să permită comunicarea bidirecțională între microcontroller și ecranul LCD. Acesta permite transferul de date între dispozitive prin intermediu două linii de comunicație. Linia SDA aceasta este linia pe care datele sunt transmise între dispozitive. Atât microcontrollerul, cât și ecranul LCD pot trimite și primi date pe această linie. Linia SCL aceasta este linia pe care semnalele de ceas sunt transmise. Semnalul de ceas este controlat de microcontroller și stabilește momentul la care datele sunt trimise sau citite.

Termocuplul tip K are două fire metalice legate într-un punct numit joncțiune, Această joncțiunea de măsurare este supusă unei diferențe de temperatură, apare o tensiune termoelectrică între cele două fire. Această tensiune este măsurată de către un circuit de măsurare și este folosit pentru măsurarea temperaturi.

Modulul MAX6675 este un circuit integrat care servește drept convertor termoelectric pentru termocupluri tip K. Acest modul preia tensiunea termoelectrică generată de joncțiunea termocuplului tip K și o convertește într-o valoare digitală reprezentând temperatura.

MAX6675 este conectat la pinii digital de la arduino pentru a transmite datele de la termocuplu. Pin-ul SCK este utilizat pentru sincronizarea transferului de date în modul serial. Pin-ul CS este utilizat

pentru a activa sau dezactiva MAX6675. Când CS este activat (adesea printr-o valoare LOW pe pin), MAX6675 va începe să convertească datele termocuplului și va trimite datele către microcontroller. Când CS este dezactivat (de obicei, printr-o valoare HIGH pe pin), MAX6675 intră într-un mod de repaus și nu transmite date. Pin-ul SO reprezintă ieșirea serială a datelor de temperatură convertite de MAX6675.

Modulul MAX6675 care este conectat SCK la pinul D7, CS la pinul D6 și SO la pinul D5.

4.2 Setarea controllerului PID

Constantele k_p , k_i , k_d , din cod sunt parametri utilizați în controllerul PID. Acești parametri determină comportamentul controllerului PID.

Câștigul proporțional (K_P) determină răspunsul controllerului la diferența dintre valoarea de referință și temperatura curentă. Este componenta principală a controllerului PID și contribuie la răspunsul imediat la modificările erorii. O valoare mare a câștigului proporțional are ca rezultat un răspuns mai puternic la eroare făcând sistemul să reacționeze mai agresiv la abaterile de la valoarea de referință. Cu toate acestea valorile câștigului proporțional foarte mari pot duce la oscilații mari sau instabilitate.

Câștigul integral (K_I) ține cont de eroarea acumulată de-a lungul timpului și asigură la eliminarea erorilor de stare staționară care nu pot fi corectate doar de componenta proporțională. O valoare mai mare a câștigului integral mărește efectul erorilor din trecut, ajutând controllerul să elimine orice eroare rămasă la starea de echilibru. Cu toate acestea valori prea mari ale câștigului integral pot duce la depășire.

Câștigul derivat (K_D) este utilizat pentru a prezice tendința viitoare a erorii pe baza ratei actuale de schimbare. Atenuază răspunsul la schimbările rapide ale erorii și ajută la prevenirea depășirilor și oscilațiilor. O valoare mai mare a câștigului derivat adaugă mai multă amortizare sistemului, reducând răspunsul la modificări rapide ale erorii. Cu toate acestea, valori foarte mari a câștigului derivat pot face controlul prea lent pentru a reacționa la schimbări bruște.

Pentru a seta valorile implicite pentru PID se începe cu valori de la zero la toate constantele. Nu putem vedea un semnal al rezistenței, dar putem vedea temperatura în timp real cu ajutorul termocuplului. Deci vom crește treptat valoarea la fiecare constantă pe rând și vom observa schimbările temperaturii.

Prima constantă pe care încep să o modific este k_p , aceasta este și cea mai importantă. Încep să măresc valoarea k_p până văd că începe să aibă un comportament mai apropiat de valoare setată, dacă încălzirea este prea lentă valoare trebuie mărită, dacă temperatura trece mult peste setpoint înainte să se stabilizeze, valoarea trebuie scăzută. Câștigul proporțional trebuie să aibă o valoare la care să ofere un răspuns rapid, dar stabil al sistemului. Se ajustează valoarea k_p până când observăm un comportament mai apropiat de valoarea setată.

A doua constantă pe care o să o modific este k_i . Observăm dacă există eroare persistentă, adică dacă sistemul nu ajunge exact la setpoint și rămâne cu o diferență mică de temperatură. Dacă există eroare persistentă vom începe prin a mării k_i treptat. Vom continua să mărim valoarea k_i până când eroarea persistentă dispare sau când valoarea temperaturii începe să oscileze excesiv în jurul setpoint-ului. Câștigul integral trebuie să aibă o valoare la care să asigure o corecție fină și să aducă sistemul cât mai aproape de setpoint.

Ultima constantă pe care o să o modific este k_d . Observăm dacă sistemul începe să oscileze excesiv sau să aibe o supracorecție este util să o folosim k_d pentru amortizare. Se mărește constant până când oscilațiile se reduc. Câștigul derivat trebuie să aibe o valoare la care să reducă oscilațiile și să aducă sistemul la un control stabil al temperaturii.

Această metodă de a găsi valorile controllerului PID nu este foarte precisă, dar se poate obține un controller PID decent pentru controlul temperaturii.

În cod se poate observa că valoarea de întârziere maximă după detectarea zero este 7400. Am ales această valoare deoarece la priză avem o frecvență de 50Hz, perioada fiind 20ms. Controlând întârzierea de tragere la fiecare jumătate de perioadă deci la fiecare 10ms. Pentru a fi sigur că nu vom trece de cei 10ms am ales valoare 7400 și deoarece dacă setăm o valoare mai aproape de 10ms temperatura creștea fără să țină cond de constantele PID.

4.3 Asamblarea circuitului

Schema tehnică a circuitului unde am folosit programul KiCad 7.0 va fi în anexă împreună cu schematicul circuitului imprimat.

În programul dedicat KiCad am avut opțiunea de a putea realiza schema tehnică folosind subprogramul Schematic Editor, acolo am făcut conexiunile pentru fiecare componentă. După conectarea componentelor am început să atribui amprenta pentru fiecare componentă am făcut asta cu ajutorul bibliotecii de la JLCPCB pentru a putea vedea ce au în stoc. Această companie chineză are o gamă largă de componente și chiar pot să ajute la printarea circuitului.

Pentru proiectarea plăcuței am folosit subprogramul de la KiCad PCB Editor. În acest program am făcut traseul pentru fiecare componentă cât și găurile pentru fiecare pin al componente ba chiar și găuri pentru montaj.

În program am accesat la o gamă largă de amprente acesta fiind și motivul pentru care am ales programul KiCad, dar nu la fiecare componentă am avut amprentă ca de exemplu redresorul în punte KBPC1010, pentru redresor am folosit opțiunea de Footprint Editor pentru a crea amprenta folosindu-mă de fișa tehnică unde am avut dimensiunea componentei.

La final am folosit lista de materiale pentru schemă pentru a ști exact ce componentă o să am nevoie la printarea circuitului și ce referință o să aibe. La fel am creat lista pentru plasarea componentelor, acesta are rol de a ști exact pe ce suprafață vine plasată, poziția exactă cât și unghiul de rotație a componentei. A mai fost creată și fișierul gerber acesta conține toate datele PCB-ului, găurile care urmează să fie făcute cât și locația lor, la fel conține fiecare strat al PCB-ului de la traseul cuprului cât și silkscreen-ul.

Pentru testarea circuitului am folosit o plăcuță tip breadboard unde am conectat toate componentele. Acestea au fost montate pe plăcuța breadboard și conectate între ele cu ajutorul sârmelor de cupru cu izolație. Modulul MAX6675 cu termocuplu cât și modulul LCD cu I2C nu sunt montate pe breadboard dar sunt conectate cu ajutorul firelor dupont. Se poate vedea în Figura 28.

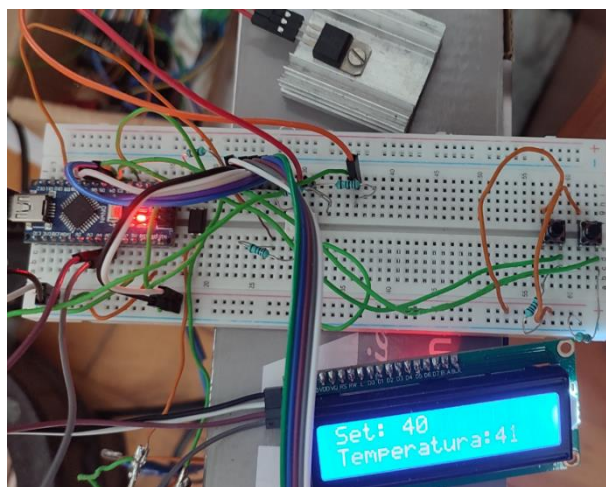


Figura 28. Montarea componentelor pe o plăcuță tip breadboard

Măsurarea temperaturii este făcută cu ajutorul modului MAX6675 și termocuplui tip K. Acesta a venit cu un termocuplu standard, dar am ales să îl schimb deoarece răspunsul la citirea temperaturii era foarte lent făcând controlul dificil se poate vedea în Figura 29.

Termocuplul pe care l-am ales se poate vedea în Figura 30. Acesta este tot un termocuplu tip K, dar este mult mai sensibil făcând citirea temperaturii mai repede ajutând să facă corecția lui într-un timp mai scurt.



Figura 29. Termocuplu tip K standart

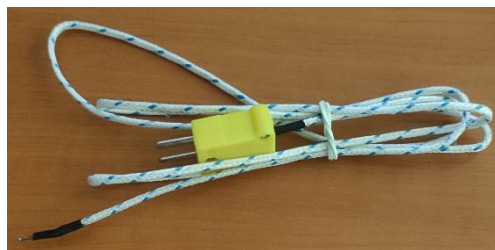


Figura 30. Termocuplu tip K folosit

Ca și element de încălzire pentru testare am folosit un reșou electric se poate vedea în Figura 31. Acesta este un simplu reșou electric model destul de vechi după cum se vede din 1975, ca și specificații tehnice are 600W și funcționează la 220V. Acesta are rezistențe ca și element de încălzire înconjurat cu samotă ceramică.



Figura 31. Reșou electric folosit pentru

Termocuplul este lăsat pe reșou să poată măsura în timp real temperatura, astfel controllerul PID să poate să acționeze mai repede pentru controlul temperaturii.

În cazul acestui reșou putem pune termocuplu în două locații. O locației ar fii direct pe rezistența lui între șamota ceramică, în acest caz putem citi temperatura directă a rezistenței și putem să acționăm rapid la schimbările de temperatură a acesteia, dar dezavantajul este că temperatura rezistenței nu este la fel dacă am măsura temperatura deasupra ei de exemplu pe șamota ceramică. A doua locației este să punem termocuplu direct pe șamota ceramică, aceasta se încălzește mai greu, dar și reține căldură făcând controlul temperaturii și schimbarea ei foarte lentă. Controlul temperaturii de pe șamotă ne interesează dacă vrem să îl folosim în practică și pentru a face acest lucru folosim funcția media mobilă.

Medie mobilă joacă un rol important în stabilizarea citirilor temperaturii și în reducerea fluctuațiilor care pot să apară în citirile senzorului.

Motivul folosirii lui este de a evita oscilațiile controlul PID poate oscila sau reacționa excesiv la fluctuațiile rapide ale citirilor. Media mobilă atenuează aceste fluctuații, ajutând la evitarea reacțiilor nedorite ale controlului. Un alt motiv este pentru răspuns lent al controlului făcând media a 5 citiri de temperatură va încetini răspunsul citiri temperaturii dar nu și reacția ei, astfel va îmbunătăți și precizia.

4.4 Testarea

Urmează setarea constantelor de câștig PID. Aceste constate sunt adăugate direct în cod unde împreună cu valoarea erorii PID care este diferența dintre setpoint și temperatura reală vor calcula valorile pentru P (proporțional), I (integral) și D (derivativ) valoarea acestora adunate va da valoarea finală a PID care este comparată cu perioada de la tensiunea din priză pentru a face o întârziere, această întârziere va afecta tiristorul care va închide și deschide circuitul la nevoie.

Pentru că nu am la dispoziție o modalitate de a vedea semnalul de ieșire a rezistenței reșoului va trebui să fac multiple testări și măsurători pentru a afla valori aproximative pentru constante de câștig PID.

Voi aplica metologia explicată mai sus la subtitlul Setarea controllerului PID. Voi începe prin a seta un setpoint la o valoare mică cum a fii 40°C, deoarece va fii o diferență de temperatură mică face de cea din cameră, acest lucru mă va ajuta să observ stabilitatea controllerului, dacă temperatura va sării cu mult setpoint-ul înseamnă că nu este stabil deloc. După care voi schimba valoare setpoint-ului, aici o pot face în două metode să o schimb lent, făcând asta nu o să arate o oscilație mare, sau pot să o măresc repede, făcând asta temperatura va oscila mai mult în jurul setpoint-ului având tendința de a depăși setpoint-ul.

Voi începe prin a lăsa toate valorile la zero și mă voi ocupa doar de câștigul proporțional. Am să stau atent să urmăresc schimbarea temperaturii și pentru a testa să văd cât de aproape sunt de valoarea corectă. Am să schimb setpoint-ul cu ajutorul butoanelor pentru a vedea cât de mult oscilează temperatura în jurul setpoint-ului. Voi începe să măresc valoarea constantei în cod puțin câte puțin și când văd ca valoare temperaturii oscilează cât mai puțin înseamnă că mă apropi de o valoare bună.

Dacă o să văd că temperatura rămâne constantă dar nu încă nu ajunge la setpoint înseamnă că o să avem nevoie de câștig integral. Acesta are rol de a elimina eroarea staționară.

Din nou voi începe de la o valoare mică, ca de exemplu 0.1 și voi urmări să văd dacă se elimină eroarea staționară. Dacă văd ca valoare staționară dispăre adică crește la valoarea setpoint-ului înseamnă că mă apropi de valoarea dorită. Dacă depășesc setpoint-ul și rămâne cu oscilație înseamnă că am o valoare mult prea mare. Dacă temperatura o să aibe o supracorecție atunci vom folosi câștigul derivat.

Din nou o să încep să facem teste dar de data asta pentru k_d câștigul derivat, vom începe de la o valoare mică și vom continua să o creștem valoare puțin câte puțin, până vedem că se menține la temperatura setată.

Evident că putem să revenim la oricare dinte valorile de câștig PID dacă credem că nu a fost setat corect.

Reglarea acestor valori se poate reface de câte ori este nevoie până o să avem un control satisfăcător al temperaturii.

Valorile finale de testarea reșoul sunt $k_p = 79$; $k_i = 0.8$; $k_d = 0.45$;

Concluzie

Valorile finale nu sunt ideale ne având posibilitatea de a vedea semnalul de ieșire a elementului de încălzire, sunt doar niște valori aproximative, evident cu cât se fac mai multe testări cu atât poți să obții valori cât mai bune rezultând într-un control mai fin și mai precis.

În testare am folosit reșoul electric despre care am discutat anterior și este prezent în Figura 30 și am reușit să controlez temperatura acestuia cu o precizie destul de bună ,chiar și șamota ceramică pe el.

De ținut minte aceste valori sunt bune doar pentru acest tip de element de încălzire adică reșoul electric folosit. Dacă elementul de încălzire va fii schimbat valorile câștigului nu vor fii corecte și nu o să funcționeze corect. Deci este obligatoriu să se facă testari sau simulări pentru noi valori de câștig PID.

Bibliografie

- [1] Niculaie Palaghiță, Dorin Petreus, Cristian Fărcaș, *Electronică de comandă și reglaj*, editura Mediamira Cluj-Napoca 2006.
- [2] Ionuț Ciocan, Niculaie Palaghiță, Dorin Petreus, Cristian Fărcaș, *Electronică de putere*, editura Risoprint Cluj-Napoca 2017.
- [3] <https://www.scritub.com/tehnica-mecanica/Regulatoare-automate44397.php>
- [4] http://www.euedia.tuiasi.ro/uee/uee_files/Lucrarea%206.pdf
- [5] <https://uk.rs-online.com/web/content/discovery/ideas-and-advice/thermocouples-guide>
- [6] <https://ro.wikipedia.org/wiki/Termocuplu>
- [7] <https://tempsens.com/blog/k-type-thermocouple>
- [8] <https://www.analog.com/media/en/technical-documentation/data-sheets/MAX6675.pdf>
- [9] https://en.wikipedia.org/wiki/Arduino_Nano
- [10] https://www.mouser.com/pdfdocs/Gravitech_Arduino_Nano3_0.pdf
- [11] http://handsontec.com/dataspecs/module/I2C_1602_LCD.pdf
- [12] <https://pdf1.alldatasheet.com/datasheet-pdf/view/22039/STMICROELECTRONICS/BTA16.html>
- [13] <https://www.alldatasheet.com/datasheet-pdf/pdf/5038/MOTOROLA/MOC3020.html>
- [14] <https://www.elprocus.com/pc817-optocoupler/>
- [15] <https://www.alldatasheet.com/datasheet-pdf/pdf/43368/SHARP/PC817.html>
- [16] <https://pdf1.alldatasheet.com/datasheet-pdf/view/810565/CHENDA/KBPC1010.html>
- [17] <https://dokumen.tips/download/link/opto-triac-moc3020-appnote.html>
- [18] https://en.wikipedia.org/wiki/Diode_bridge
- [19] <https://i.electricianexp.com/fr/main/praktika/1400-kak-proverit-diodnyy-most.html>

Anexe

Codul folosit și comentat

```
#include "max6675.h" // Include biblioteca pentru comunicarea cu senzorul MAX6675
#include <Wire.h> // Include biblioteca pentru comunicarea I2C
#include <LiquidCrystal_I2C.h> // Include biblioteca pentru controlul unui LCD I2C

int firing_pin = 9; // Pinul folosit pentru controlul tiristorului / triacului
int increase_pin = 4; // Pinul pentru butonul de creștere a setării temperaturii
int decrease_pin = 3; // Pinul pentru butonul de scădere a setării temperaturii
int zero_cross = 2; // Pinul pentru detectarea trecerii prin zero a tensiunii
int thermoDO = 5; // Pinul pentru datele de ieșire ale senzorului termocupla
int thermoCS = 6; // Pinul pentru selectarea senzorului termocupla
int thermoCLK = 7; // Pinul pentru semnalul de ceas al senzorului termocupla

MAX6675 thermocouple(thermoCLK, thermoCS, thermoDO); // Inițializează obiectul
pentru citirea temperaturii

LiquidCrystal_I2C lcd(0x27, 20, 4); // Inițializează obiectul pentru controlul LCD-
ului

bool zero_cross_detected = false; // Indicator pentru detectarea trecerii prin zero
a tensiunii
int firing_delay = 7400; // Timpul de întârziere pentru declanșarea tiristorului /
triacului
int maximum_firing_delay = 7400; // Valoarea maximă pentru timpul de întârziere

unsigned long previousMillis = 0; // Momentul în milisecunde al ultimei iterații
unsigned long currentMillis = 0; // Momentul curent în milisecunde
int temp_read_Delay = 500; // Intervalul de citire a temperaturii
int real_temperature = 0; // Valoarea temperaturii măsurate
int setpoint = 40; // Valoarea țintă pentru temperatură

bool button_increase_state = LOW; // Starea butonului de creștere a setării
bool button_decrease_state = LOW; // Starea butonului de scădere a setării

float PID_error = 0; // Eroarea PID
float previous_error = 0; // Eroarea PID anterioară
float elapsedTime, Time, timePrev; // Variabile pentru calculul timpului pentru PID
int PID_value = 0; // Valoarea calculată de PID

int kp = 79; // Coeficientul proportional al controllerului PID
int ki = 0.8; // Coeficientul integral al controllerului PID
int kd = 0.45; // Coeficientul diferențial al controllerului PID

int PID_p = 0; // Termenul proportional al controllerului PID
```

```

int PID_i = 0; // Termenul integral al controllerului PID
int PID_d = 0; // Termenul diferențial al controllerului PID

int temperatureBuffer[5]; // Buffer pentru citirile temperaturii
int bufferIndex = 0; // Indexul pentru buffer

void setup() {
    pinMode(firing_pin, OUTPUT); // Configurează pinul pentru declanșare ca ieșire
    pinMode(zero_cross, INPUT_PULLUP); // Configurează pinul pentru detectarea zero
    crossing ca intrare cu rezistență pull-up
    pinMode(increase_pin, INPUT_PULLUP); // Configurează pinul pentru butonul de
    creștere ca intrare cu rezistență pull-up
    pinMode(decrease_pin, INPUT_PULLUP); // Configurează pinul pentru butonul de
    scădere ca intrare cu rezistență pull-up

    lcd.init(); // Inițializează LCD-ul
    lcd.backlight(); // Activează iluminarea LCD-ului
}

void loop() {
    currentMillis = millis(); // Actualizează momentul curent

    if (currentMillis - previousMillis >= temp_read_Delay) {
        previousMillis += temp_read_Delay;

        // Citirea temperaturii în buffer și calculul mediei mobile
        temperatureBuffer[bufferIndex] = thermocouple.readCelsius();
        bufferIndex = (bufferIndex + 1) % 5;
        real_temperature = calculateMovingAverage();

        PID_error = setpoint - real_temperature; // Calculează eroarea PID

        if (PID_error > 20) {
            PID_i = 0; // Resetarea termenului integral dacă eroarea este mare
        }

        PID_p = kp * PID_error; // Calculează termenul proportional
        PID_i = PID_i + (ki * PID_error); // Calculează termenul integral

        // Calcularea timpului scurs între iterații pentru PID
        timePrev = Time; // Salvarea timpului precedent
        Time = millis(); // Actualizarea timpului curent
        elapsedTime = (Time - timePrev) / 1000; // Calculul timpului scurs în secunde
        PID_d = kd * ((PID_error - previous_error) / elapsedTime); // Calculul termenului
        diferențial pentru PID
        PID_value = PID_p + PID_i + PID_d; // Calculul valorii finale a controlului PID
    }
}

```



```

if (PID_value < 0) {
    PID_value = 0; // Asigurarea că valoarea PID este între 0 și maximum_firing_delay
}
if (PID_value > 7400) {
    PID_value = 7400; // Limitarea valorii PID la maximum_firing_delay
}

// Actualizarea afișajului LCD cu valorile PID și setpoint-ului
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Set: ");
lcd.setCursor(5, 0);
lcd.print(setpoint);
lcd.setCursor(0, 1);
lcd.print("Temperatura: ");
lcd.setCursor(12, 1);
lcd.print(real_temperature);
previous_error = PID_error; // Actualizarea eroarei anterioare pentru termenul
diferențial al PID
}

// Verificarea stării butonului de creștere și scădere a setării temperaturii
if (digitalRead(increase_pin) == LOW) { // Verificarea dacă butonul de creștere
este apăsător
    if (!button_increase_state) { // Verificarea dacă butonul tocmai a fost apăsător
        setpoint += 5; // Incrementarea valorii țintă a temperaturii
        button_increase_state = true; // Actualizarea stării butonului
    }
} else {
    button_increase_state = false; // Resetarea stării butonului
}

if (digitalRead(decrease_pin) == LOW) { // Verificarea dacă butonul de scădere este
apăsător
    if (!button_decrease_state) { // Verificarea dacă butonul tocmai a fost apăsător
        setpoint -= 5; // Decrementarea valorii țintă a temperaturii
        button_decrease_state = true; // Actualizarea stării butonului
    }
} else {
    button_decrease_state = false; // Resetarea stării butonului
}

// Detectarea trecerii prin zero
if (digitalRead(zero_cross) == HIGH) {
    zero_cross_detected = true; // Indică detectarea trecerii prin zero
}

```

```

// Controlul declanșării triacului
if (zero_cross_detected) {
    delayMicroseconds(maximum_firing_delay - PID_value); // Aplică întârzierea pentru
declanșare
    digitalWrite(firing_pin, HIGH); // Activează tiristorul / triacul
    delayMicroseconds(100); // Pauză mică pentru a asigura declanșarea corectă
    digitalWrite(firing_pin, LOW); // Dezactivează tiristorul / triacul
    zero_cross_detected = false; // Resetează indicatorul de trecere prin zero
}
}
// Funcția pentru calcularea mediei mobile
int calculateMovingAverage() {
    int sum = 0;
    for (int i = 0; i < 5; i++) {
        sum += temperatureBuffer[i]; // Adaugă valoarea temperaturii în suma totală
    }
    return sum / 5; // Calculează și returnează media mobilă
}

```