

# અમ્બડેડ સિસ્ટમ (4343204) - વિન્ટર 2024 સોલ્યુશન

Milav Dabgar

December 03, 2024

## પ્રશ્ન 1(a) [3 ગુણ]

ATmega32 માં RAM, Flash અને EEPROM મેમરી કેટલી છે? માઇકોકન્ટ્રોલરમાં તેની જરૂરિયાત સમજાવો.

### જવાબ

ATmega32 મેમરી સ્પેસિફિકેશન અને માઇકોકન્ટ્રોલર ઓપરેશનમાં તેનું મહત્વ:

કોષ્ટક 1. ATmega32માં મેમરી સાઇઝ

| મેમરી પ્રકાર | સાઇઝ  | હેતુ                        |
|--------------|-------|-----------------------------|
| SRAM (RAM)   | 2 KB  | વેરિએબલ્સ અને સ્ટેક સ્ટોરેજ |
| Flash        | 32 KB | પ્રોગ્રામ સ્ટોરેજ           |
| EEPROM       | 1 KB  | નોન-વોલેટાઇલ ડેટા સ્ટોરેજ   |

- RAM: પ્રોગ્રામ એક્ઝિક્યુશન દરમિયાન વેરિએબલ્સ માટે ટેમ્પરરી સ્ટોરેજ
- Flash: પ્રોગ્રામ ઇન્સ્ટ્રુક્શન્સ અને કોન્સ્ટન્ટ્સ માટે પરમેનન્ટ સ્ટોરેજ
- EEPROM: પાવર સાયકલ્સ પછી પણ જાળવી રાખવા જરૂરી એવા ડેટા માટે લાંબા ગાળાનું સ્ટોરેજ

### મેમરી ટ્રીક

"ન માટે RAM, ફંક્શન માટે Flash, હ્યુમેન માટે EEPROM"

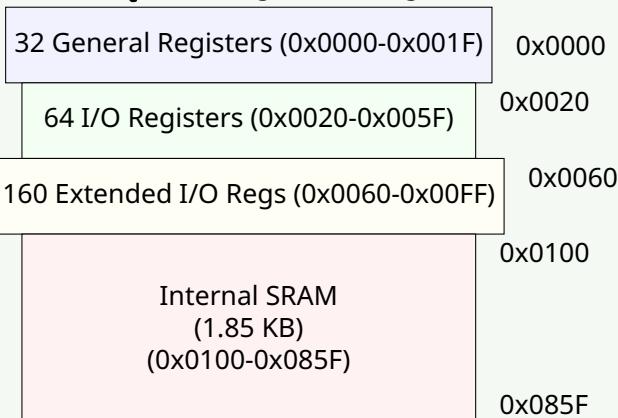
## પ્રશ્ન 1(b) [4 ગુણ]

ATmega32 ની RAM મેમરીની ચર્ચા કરો.

### જવાબ

ATmega32ની RAM (SRAM) ચોક્કસ હેતુઓ માટે જુદા જુદા વિભાગોમાં ગોઠવાયેલી છે.

આકૃતિ 1. ATmega32 RAM Organization



- રજિસ્ટર ફાઇલ: પ્રથમ 32 લોકેશન્સ (0x0000-0x001F)
- I/O રજિસ્ટર્સ: સ્ટાન્ડર્ડ I/O સ્પેસ (0x0020-0x005F)
- એક્સ્ટેન્ડેડ I/O: વધારાના પેરિફેરલ રજિસ્ટર્સ (0x0060-0x00FF)
- ડેટા મેમરી: જનરલ પરપ્ય �SRAM (0x0100-0x085F)

#### મેમરી ટ્રીક

"રજિસ્ટર્સ, I/O, એક્સ્ટેન્ડેડ, ડેટા - RAM ની કાર્યક્ષમ ડિઝાઇન"

## પ્રશ્ન 1(c) [7 ગુણ]

રિયલ ટાઈમ ઓપરેટિંગ સિસ્ટમની વ્યાખ્યાયિત કરો અને તેની લાક્ષણિકતાઓ સમજાવો.

#### જવાબ

રિયલ-ટાઈમ ઓપરેટિંગ સિસ્ટમ (RTOS) એ ચુસ્ત ટાઈમિંગ જરૂરિયાતો સાથે ડેટા અને ઇવેન્ટ્સ પ્રોસેસ કરવા માટે ડિઝાઇન કરાયેલ સ્પેશિયલાઇઝ ઓપરેટિંગ સિસ્ટમ છે.

કોષ્ટક 2. RTOS ની મુખ્ય લાક્ષણિકતાઓ

| લાક્ષણિકતા             | વર્ણન  |
|------------------------|--|
| ડિટમિનિઝમ              | ટાસ્ક્સ માટે ગેરટેડ રિસ્પોન્સ ટાઈમ   |
| પ્રિયેમિટ્વ શેડ્યુલિંગ | ઉચ્ચ પ્રાધાન્યવાળા ટાસ્ક્સ નીચા પ્રાધાન્યવાળા ટાસ્ક્સને ઇન્ટરપટ કરી શકે છે |
| લો લેટન્સી             | ઇવેન્ટ અને રિસ્પોન્સ વચ્ચે ન્યૂનતમ વિલંબ                                   |
| પ્રાથોરિટી-બેર્ડ       | એક્ઝિક્યુશન માટે ટાસ્ક્સને પ્રાધાન્ય આપવામાં આવે છે                        |
| ટાસ્ક મેનેજમેન્ટ       | ટાસ્ક કિએશન, ડિલીશન અને સિંકનાઇઝેશન માટે મેકેનિઝમ્સ પૂરા પાડે છે           |
| રિસોર્સ મેનેજમેન્ટ     | રિસોર્સ કોન્ફિલિક્ટ્સ અને ડેફ્લોક્સ અટકાવે છે                              |
| વિશ્વસનીયતા            | પીક લોડ હેઠળ પણ મજબૂત ઓપરેશન   |

- મલ્ટીટાસ્કિંગ: અનેક ટાસ્ક્સના કન્કરન્ટ એક્ઝિક્યુશનને સપોર્ટ કરે છે
- સ્પોલ ફૂટપ્રિન્ટ: મયાર્ગિદિત રિસોર્સવાળા એમ્બેડેડ સિસ્ટમ્સ માટે ઓપ્ટિમાઇઝડ
- ટાઈમ મેનેજમેન્ટ: માઇકોસેકન્ડ રેજોલ્યુશન સાથે પ્રિસાઇઝ ટાઈમિંગ સર્વિસીસ
- કર્નલ સર્વિસીસ: ટાસ્ક કોઓર્ડિનેશન માટે IPC, મ્યુટેક્સ, સેમાફોર

#### મેમરી ટ્રીક

"ડિટમિનિઝિક પ્રિયેમિટ્વ ટાસ્ક્સ નં ઓન સ્ટ્રિક્ટ ટાઈમલાઇન્સ"

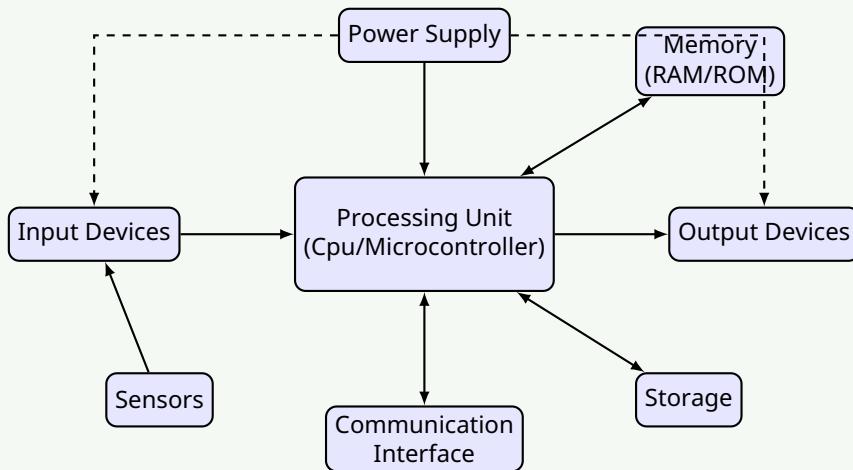
## પ્રશ્ન 1(c OR) [7 ગુણ]

એમ્બેડેડ સિસ્ટમ શું છે? એમ્બેડેડ સિસ્ટમનો સામાન્ય બ્લોક ડાયાગ્રામ દોરો અને સમજાવો.

#### જવાબ

એમ્બેડેડ સિસ્ટમ એ એક ડેડિકેટેડ કમ્પ્યુટર સિસ્ટમ છે જે મોટી ભિકેનિકલ અથવા ઇલેક્ટ્રિકલ સિસ્ટમની અંદર ચોક્કસ કાર્યો કરવા માટે ડિઝાઇન કરવામાં આવે છે, ધારીવાર રિયલ-ટાઈમ કન્સ્ટ્રેઇન્ટ્સ સાથે.

આકૃતિ 2. General Block Diagram of Embedded System



કોષ્ટક 3. એમ્બેડેડ સિસ્ટમ કોમ્પોનેન્ટ્સ

| કોમ્પોનેન્ટ      | ફંક્શન  |
|------------------|---|
| પ્રોસેસિંગ યુનિટ | પ્રોગ્રામ ઇન્સ્ટ્રક્શન્સ એક્ઝિક્યુટ કરે છે (માઇકોકન્ટ્રોલર/માઇકોપ્રોસેસર) |
| મેમરી            | પ્રોગ્રામ અને ડેટા સ્ટોર કરે છે (RAM, ROM, Flash)                         |
| ઇનપુટ/આઉટપુટ     | બાહ્ય ડિવાઇસ સાથે ઇન્ટરફેસ કરે છે   |
| કમ્યુનિકેશન      | અન્ય સિસ્ટમ્સ અથવા નેટવર્કર્સ સાથે જોડાય છે                               |
| પાવર સાખાય       | રેચ્યુલેટેડ પાવર પ્રદાન કરે છે  |
| સેન્સર્સ         | પર્યાવરણીય ડેટા એક્તિત કરે છે   |

- એપ્લિકેશન-સ્પેચિફિક: ડેડિકેટેડ ટાસ્ક્સ માટે ડિજાઇન કરાયેલ
- રિસોર્સ-કન્સ્ટ્રેઇન્ડ: મર્યાદિત પ્રોસેસિંગ પાવર અને મેમરી
- રિયલ-ટાઈમ: ટાઈમિંગ કન્સ્ટ્રેઇન્સની અંદર ઇવેન્ટ્સને પ્રતિસાદ આપે છે
- હાઇ રિલાયબિલિટી: નિષ્ફળતા વિના સતત ઓપરેટ કરવું જોઈએ

### મેમરી ટ્રીક

“પ્રોસેસ, મેમરી, I/O - દરેક સિસ્ટમમાં હોવું જોઈએ”

## પ્રશ્ન 2(a) [3 ગુણ]

એમ્બેડેડ સિસ્ટમમાં કોઈપણ એપ્લિકેશન ડિજાઇન માટે માઇકોકન્ટ્રોલર પસંદ કરવા માટે વિવિધ માપદંડો લખો.

### જવાબ

યોગ્ય માઇકોકન્ટ્રોલર પસંદ કરવા માટે એપ્લિકેશન જરૂરિયાતો આધારિત અનેક માપદંડોનું મૂલ્યાંકન કરવું જરૂરી છે.

કોષ્ટક 4. માઇકોકન્ટ્રોલર પસંદગી માપદંડ

| માપદંડ          | વિચારણાઓ                               |
|-----------------|--|
| પરફોર્મન્સ      | CPU સ્પીડ, MIPS, બિટ વિડ્યુલ (8/16/32) |
| મેમરી           | Flash, RAM, EEPROM ડેપેસિટી            |
| પાવર કન્જામ્પશન | ઓપરેટિંગ વોલ્ટેજ, સ્લીપ મોડ            |
| I/O કેપેબિલિટી  | પોર્ટ્સની સંખ્યા, સ્પેશિયલ ફંક્શન્સ    |
| પેરિફેરલ્સ      | ADC, ટાઇમર્સ, કમ્યુનિકેશન ઇન્ટરફેસીસ   |
| કોર્સ           | યુનિટ પ્રાઇસ, ડેવલપમેન્ટ ટૂલ્સ         |
| ફોર્મ ફેક્ટર    | સાઇઝ, પેકેજ ટાઇપ, પિન કાઉન્ટ           |

- એપ્લિકેશન રિકવાયરમેન્ટ્સ: એપ્લિકેશન માટે જરૂરી સ્પેચિફિક ફીચર્સ
- ડેવલપમેન્ટ એન્વાયરન્મેન્ટ: ઉપલબ્ધ કમ્પાઇલર્સ, ડિબગર્સ, લાઇસેરીઝ
- ફ્યુચર એક્સપાન્શન: ભવિષ્યના એન્હાન્સમેન્ટ્સ માટે સ્કેલેબિલિટી

## મેમરી ટ્રીક

“પરફોર્મન્સ મેમરી પાવર I/O કોસ્ટ”

## પ્રશ્ન 2(b) [4 ગુણ]

TCCRO રજિસ્ટર દોરો અને સમજાવો.

## જવાબ

ટાઇમર/કાઉન્ટર કંટ્રોલ રજિસ્ટર 0 (TCCRO) ATmega32માં ટાઇમર/કાઉન્ટરના ઓપરેશનને કંટ્રોલ કરે છે.

આનુક્તિક 3. TCCRO Register

|           |            |            |            |            |           |           |           |
|-----------|------------|------------|------------|------------|-----------|-----------|-----------|
| FOC0<br>7 | WGM00<br>6 | COM01<br>5 | COM00<br>4 | WGM01<br>3 | CS02<br>2 | CS01<br>1 | CS00<br>0 |
|-----------|------------|------------|------------|------------|-----------|-----------|-----------|

કોષ્ટક 5. TCCRO બિટ ફંક્શન્સ

| બિટસ  | નામ     | ફંક્શન                    |
|-------|---------|---------------------------|
| 7     | FOC0    | ફોર્સ આઉટપુટ કમ્પેર       |
| 6,3   | WGM01:0 | વેવફોર્મ જનરેશન મોડ       |
| 5,4   | COM01:0 | કમ્પેર મેચ આઉટપુટ મોડ     |
| 2,1,0 | CS02:0  | કલોક સિલેક્ટ (પ્રીસ્કેલર) |

- WGM01:0: ટાઇમર ઓપરેટિંગ મોડ નક્કી કરે છે (નોર્મલ, CTC, PWM)
- COM01:0: OC0 પિન આઉટપુટ બિહેવિયર કંટ્રોલ કરે છે
- CS02:0: કલોક સોર્સ અને પ્રીસ્કેલર વેવ્યુ પસંદ કરે છે

## મેમરી ટ્રીક

“ફોર્સ વેવફોર્મ કમ્પેર કલોક સિલેક્ટ”

## પ્રશ્ન 2(c) [7 ગુણ]

ATmega32 ના ટાઇમરોની યાદી બનાવો અને કોઈપણ એક ટાઇમરના Modes ને વિગતવાર સમજાવો.

## જવાબ

ATmega32માં વિવિધ ક્ષમતાઓ અને ઓપરેટિંગ મોડ્સ સાથે અનેક ટાઇમર્સ છે.

કોષ્ટક 6. ATmega32માં ટાઇમર્સ

| ટાઇમર  | પ્રકાર    | સાઇઝ   | ફીર્ચર્સ                              |
|--------|-----------|--------|---------------------------------------|
| ટાઇમર0 | જનરલ પરપા | 8-બિટ  | સિમ્પલ ટાઇમિંગ, PWM                   |
| ટાઇમર1 | એડવાન્ડ   | 16-બિટ | ઇનપુટ કેપ્ચર, ડાયલાય પ્રોસ્સેન્સ, PWM |
| ટાઇમર2 | જનરલ પરપા | 8-બિટ  | એસિંકોન્સ ઓપરેશન                      |

ટાઇમર0 ઓપરેટિંગ મોડ્સ:

- નોર્મલ મોડ:
  - કાઉન્ટર 0 થી 255 સુધી વધે છે પછી 0 પર ઓવરફ્લો થાય છે
  - ઓવરફ્લો ઇન્ટરપટ જનરેટ થઈ શકે છે
  - સરળ ટાઇમિંગ અને ડિલે જનરેશન માટે વપરાય છે
- CTC (કિલ્યુર ટાઇમર ઓન કમ્પેર) મોડ:
  - કાઉન્ટર OCRO વેવ્યુ પર પહોંચે ત્યારે રીસેટ થાય છે
  - પ્રિસાઇઝ ફિકવન્સી જનરેશન માટે ઉપયોગી

- કમ્પેર મેચ ઇન્ટરપ્ટ જનરેટ થઈ શકે છે
- ફાસ્ટ PWM મોડ:
  - કાઉન્ટર 0 થી 255 સુધી ગણો છે
  - આઉટપુટ ઓવરફલો અને કમ્પેર મેચ પર ટોગલ થાય છે
  - હાઇ ફિક્વન્સી PWM જનરેશન
- ફેઝ કરેક્ટ PWM મોડ:
  - કાઉન્ટર ઉપર પછી નીચે (0 → 255 → 0) ગણો છે
  - સિમેટ્રિક PWM વેવફોર્મ જનરેશન
  - ફાસ્ટ PWM કરતાં ઓછી ફિક્વન્સી પણ વધુ સારી રેજોલ્યુશન

### મેમરી ટ્રીક

"નોર્મલ કમ્પોર્સ ફાસ્ટ ફેઝ - ટાઇમર મોડુસ મેરર"

## પ્રશ્ન 2(a OR) [3 ગુણ]

વિવિધ એમ્બેડેડ સિસ્ટમ એપ્લિકેશન્સની સૂચિ બનાવો. કોઈપણ એકને ટૂકમાં સમજાવો.

### જવાબ

એમ્બેડેડ સિસ્ટમ્સ વિવિધ ડોમેઇન્સમાં અનેક એપ્લિકેશન્સમાં જોવા મળે છે.

કોષ્ટક 7. એમ્બેડેડ સિસ્ટમ એપ્લિકેશન્સ

| ડોમેઇન          | એપ્લિકેશન્સ   |
|-----------------|---|
| કન્યુમર         | સ્માર્ટ એપ્લાયન્સીસ, એન્ટરટેઇનમેન્ટ સિસ્ટમ્સ        |
| ઓટોમોટિવ        | એન્જિન કંટ્રોલ, સેફ્ટી સિસ્ટમ્સ, ઇન્ફોટેઇનમેન્ટ     |
| ઇન્ડસ્ટ્રિયલ    | પ્રોસેસ કંટ્રોલ, ઓટોમેશન, રોબોટિક્સ                 |
| મેડિકલ          | પેશાન્ત મોનિટરિંગ, ઇમેજિન્ગ, ઇમ્પ્લાન્ટેબલ ડિવાઇસીસ |
| કમ્પ્યુનિકેશન્સ | રાઉટર્સ, મોડેમ્સ, નેટવર્ક સ્વિચીસ                   |
| એરોસ્પેસ        | ફલાઇટ કંટ્રોલ, નેવિગેશન, લાઇફ સપોર્ટ                |

સ્માર્ટ હોમ ઓટોમેશન સિસ્ટમ: સ્માર્ટ હોમ સિસ્ટમ ઘરેલું ઉપકરણોને મોનિટર અને કંટ્રોલ કરવા માટે એમ્બેડેડ કન્ટ્રોલર્સનો ઉપયોગ કરે છે. સેન્સર્સ તાપમાન અને મોશન જેવી પર્યાવરણીય સ્થિતિઓને ડિટેક્ટ કરે છે, જ્યારે માઇકોકન્ટ્રોલર્સ આ ડેટાને પ્રોસેસ કરે છે અને HVAC સિસ્ટમ્સ, લાઇટિંગ અને સિક્યુરિટી ડિવાઇસીસ જેવા એકચ્ચુએટર્સને કંટ્રોલ કરે છે. સિસ્ટમને ઓટોનોમિસ ઓપરેશન અથવા સ્માર્ટફોન અપ્સ દ્વારા યુઝર કંટ્રોલ માટે પ્રોગ્રામ કરી શકાય છે, જે સુવિધા, એનર્જી એફિષિયન્સી અને એન્હાન્ડ સિક્યુરિટી પ્રદાન કરે છે.

### મેમરી ટ્રીક

"કન્યુમર્સ ઓટોમેટ ઇન્ડસ્ટ્રી મેડિકલ કમ્પ્યુનિકેશન્સ એરોસ્પેસ"

## પ્રશ્ન 2(b OR) [4 ગુણ]

ATmega32 માઇકોકન્ટ્રોલરમાં DDRA, PINA અને PORTA રજિસ્ટરનાં કાર્ય સમજાવો.

### જવાબ

ત્રણ રજિસ્ટર્સ ATmega32માં પોર્ટ A ના ઓપરેશનને કંટ્રોલ કરે છે, દરેક અલગ હેતુ ધરાવે છે.

કોષ્ટક 8. પોર્ટ A રજિસ્ટર્સ

| રજિસ્ટર | ફુંક્શન           | ઓપરેશન  |
|---------|-------------------|---|
| DDRA    | ડેટા ડિરેક્શન     | પિનસને ઇનપુટ (0) અથવા આઉટપુટ (1) તરીકે કન્ફિગર કરે છે |
| PORTA   | ડેટા રજિસ્ટર      | આઉટપુટ વેલ્યુ સેટ કરે છે અથવા પુલ-અપ્સ એનેબલ કરે છે   |
| PINA    | પોર્ટ ઇનપુટ પિન્સ | એક્સ્યુઅલ પિન સ્ટેટ્સ વાંચો છે                        |

કન્ફિગરેશન ઉદાહરણો:

#### Code

```

1 DDRA = 0xFF; // બધી પનિસ આઉટપુટ તરીકે
2 PORTA = 0xA5; // આલ્ટરનેટિભ પેટ્સન સેટ કરો (10100101)
3
4 DDRA = 0x00; // બધી પનિસ ઇનપુટ તરીકે
5 PORTA = 0xFF; // બધી પનિસ પર ઇન્ટરનલ પુલઅપ્સ- એનેબલ કરો
6 data = PINA; // કર્ટ પનિ સ્ટેટ્સ વાંચો

```

- બિટ-લેવલ કંટ્રોલ: દરેક બિટ સંબંધિત પિનને કંટ્રોલ કરે છે
- એટોમિક ઓપરેશન્સ: વ્યક્તિગત બિટ્સ મોડિફાય કરી શકાય છે
- રીડ-મોડિફાય-રાઇટ: સામાન્ય ઓપરેશન પેટર્ન

#### મેમરી ટ્રીક

“ડિરેક્શન ડિટરમાઇન્સ, પોર્ટ પ્રોવાઇડ્સ, PIN પર્સોનિયલ્સ”

## પ્રશ્ન 2(c OR) [7 ગુણ]

ATmega32 નું સ્ટેટ્સ રજિસ્ટર દોરો અને તેને વિગતવાર સમજાવો.

#### જવાબ

ATmega32માં સ્ટેટ્સ રજિસ્ટર (SREG) એરિથમેટિક ઓપરેશન-સથી પ્રભાવિત પ્રોસેસર સ્ટેટ્સ ફ્લેગ્સ ધરાવે છે અને ઇન્ટરપ્ટ્સને કંટ્રોલ કરે છે.

આકૃતિ 4. Status Register (SREG)

| I | T | H | S | V | N | Z | C |
|---|---|---|---|---|---|---|---|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

કોષ્ટક 9. SREG બિટ ફુંક્શન્સ

| બિટ | નામ | ફુંક્શન                   | સેટ થાય ત્યારે                                 |
|-----|-----|---------------------------|--|
| 7   | I   | ગ્લોબલ ઇન્ટરપ્ટ એનેબલ     | પ્રોગ્રામેટિકલી એનેબલ                          |
| 6   | T   | બિટ કોપી સ્ટોરેજ          | બિટ કોપી ઇન્સ્ટ્રક્શન્સ માટે ઉપયોગમાં લેવાય છે |
| 5   | H   | હાફ ડેરી ફ્લેગ            | BCD ઓપરેશન-સમાં હાફ-ડેરી                       |
| 4   | S   | સાઇન ફ્લેગ                | $N \oplus V$ (સાઇન ઓપરેશન-સમાં માટે ઉપયોગી)    |
| 3   | V   | ટુ'સ કોમ્પિલમેન્ટ ઓવરફ્લો | એરિથમેટિક ઓવરફ્લો થાય ત્યારે                   |
| 2   | N   | નેગેટિવ ફ્લેગ             | પરિણામ નેગેટિવ છે (MSB=1)                      |
| 1   | Z   | ઝીરો ફ્લેગ                | પરિણામ ઝીરો છે                                 |
| 0   | C   | કેરી ફ્લેગ                | એરિથમેટિકમાં કેરી થાય છે                       |

- એરિથમેટિક ફીડબેક: રિજલ્ટ સ્ટેટ્સ દર્શાવે છે
- કન્ફિગરનલ બ્રાન્ચીસ: બ્રાન્ચ ઇન્સ્ટ્રક્શન્સ દ્વારા ઉપયોગ કરાય છે
- ઇન્ટરપ્ટ કંટ્રોલ: I-બિટ બધા ઇન્ટરપ્ટ્સને એનેબલ/ડિસેબલ કરે છે
- એક્સ્સેસ મેથ્ડ્સ: IN/OUT ઇન્સ્ટ્રક્શન્સ દ્વારા ડાયરેક્ટલી એડ્રેસેબલ

## મેમરી ટ્રીક

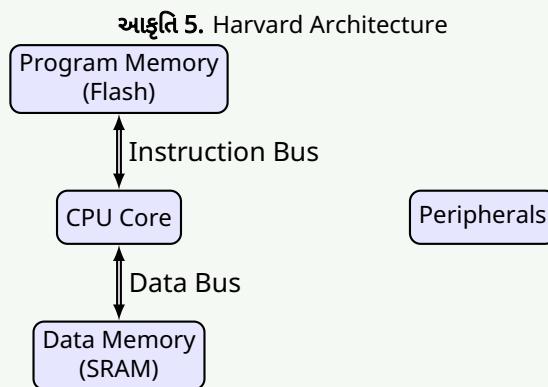
“ઇન્ટરપ્ટ્સ ટ્રેક હાફ સાઇન ઓવરફલો નેગેટિવ ઝિરો કેરી”

## પ્રશ્ન 3(a) [3 ગુણ]

AVR માઇકોકન્ટ્રોલરના હાર્વર્ડ આર્કિટેક્ચર પર ટૂંકી નોંધ લખો.

## જવાબ

હાર્વર્ડ આર્કિટેક્ચર એ AVR માઇકોકન્ટ્રોલરનો ફન્ડામેન્ટલ ડિઝાઇન પ્રિન્સિપલ છે, જે પ્રોગ્રામ અને ડેટા મેમરીને અલગ કરે છે.



- સેપરેટ બસ: પ્રોગ્રામ અને ડેટા મેમરી માટે ઇન્ડિપેન્ડન્ટ બસ
- પેરેલ એક્સેસ: એક સાથે ઇન્સ્ટ્રક્શન્સ ફૂચ અને ડેટા એક્સેસ કરી શકે છે
- પરફોર્મન્સ: મેમરી બોટલનેક્સ દૂર કરીને એક્ઝિક્યુશન સ્પીડ વધારે છે
- ડિફરન્ટ વિદ્યુસ: પ્રોગ્રામ મેમરી 16-બિટ વર્ડ્સમાં, ડેટા મેમરી 8-બિટ બાઇટ્સમાં ઓર્ગાનાઇઝ છે

## મેમરી ટ્રીક

“પ્રોગ્રામ અને ડેટા પાથ્સ અલગ છે”

## પ્રશ્ન 3(b) [4 ગુણ]

સીરીયલ કોમ્યુનિકેશન (RS232) સાથે સંકળાયેલ રજીસ્ટરોની યાદી બનાવો અને તેને ATmega32 સાથે ઇન્ટરફેસ કરવાનાં પગલાં સમજાવો.

## જવાબ

ATmega32 સીરીયલ કોમ્યુનિકેશન માટે USART (યુનિવર્સલ સિંક્રોનસ એસિંક્રોનસ રિસીવર ટ્રાન્સમિટર) નો ઉપયોગ કરે છે.

કોષ્ટક 10. USART રજિસ્ટર્સ

| રજિસ્ટર     | ફંક્શન                                |
|-------------|---------------------------------------|
| UDR         | USART ડેટા રજિસ્ટર (ટ્રાન્સમિટ/રિસીવ) |
| UCSRA       | USART કંટ્રોલ અને સ્ટેટ્સ રજિસ્ટર A   |
| UCSRB       | USART કંટ્રોલ અને સ્ટેટ્સ રજિસ્ટર B   |
| UCSRC       | USART કંટ્રોલ અને સ્ટેટ્સ રજિસ્ટર C   |
| UBRRH/UBRRL | USART બોડ રેટ રજિસ્ટર્સ               |

RS232 ઇન્ટરફેસ કરવાના પગલાં:

1. હાર્ડવેર કનેક્શન:

- ATmega32ના TXD (PD1) અને RXD (PD0) MAX232 સાથે કનેક્ટ કરો
- MAX232ને RS232 પાર્ટ અથવા કનેક્ટર સાથે કનેક્ટ કરો

## 2. USART ઇનિશિયલાઇઝ:

- બોડ રેટ સેટ કરો (UBRR)
- ફેમ ફોર્મેટ સેટ કરો (ડેટા બિટ્સ, પેરિટી, સ્ટોપ બિટ્સ)
- ટ્રાન્સમિટર અને/અથવા રિસીવર એનેબલ કરો

## 3. ડેટા ટ્રાન્સમિશન/રિસેપ્શન:

- ઓપરેશન પહેલાં સ્ટેટ્સ ફ્લેગ્સ ચેક કરો
- ટ્રાન્સમિટ કરવા માટે UDRમાં લખો
- રિસીવ કરવા માટે UDRમાંથી વાંચો

### મેમરી ટ્રીક

"કનેક્ટ, બોડ કન્ફિગર, એનેબલ, ટ્રાન્સમિટ/રિસીવ"

## પ્રશ્ન 3(c) [7 ગુણ]

જરૂરી ઉદાહરણો સાથે AVR C પ્રોગ્રામ્સમાં Bit-wise logical operations વિગતવાર ચર્ચા કરો.

### જવાબ

બિટ-વાઇઝ ઓપરેશન્સ બાઇટ અથવા વર્ડમાં વ્યક્તિગત બિટ્સને મેનિયુલેટ કરે છે, જે એમ્બેડેડ પ્રોગ્રામ્સ માટે અનિવાર્ય છે.

કોષ્ટક 11. AVR C માં બિટ-વાઇઝ ઓપરેટર્સ

| ઓપરેટર | ઓપરેશન      | ઉદાહરણ      | પરિણામ |
|--------|-------------|-------------|--------|
| &      | AND         | 0xA5 & 0x0F | 0x05   |
|        | OR          | 0x50   0x0F | 0x5F   |
| ^      | XOR         | 0x55 ^ 0xFF | 0xAA   |
| ~      | NOT         | ~0x55       | 0xAA   |
| <<     | લેફ્ટ શિફ્ટ | 0x01 << 3   | 0x08   |
| >>     | રાઇટ શિફ્ટ  | 0x80 >> 3   | 0x10   |

ઉદાહરણ: બિટ્સ સેટ અને ક્લિયર કરવી

### Code

```

1 // PORTB ની બટિ 3 સેટ કરો
2 PORTB |= (1 << 3); // PORTB = PORTB | 0b00001000
3
4 // PORTB ની બટિ 5 ફ્લાયર કરો
5 PORTB &= ~(1 << 5); // PORTB = PORTB & 0b11011111
6
7 // PORTB ની બટિ 2 ટોગલ કરો
8 PORTB ^= (1 << 2); // PORTB = PORTB ^ 0b00000100
9
10 // ચેક કરો કે બટિ 4 સેટ છે કે નહીં
11 if (PINB & (1 << 4)) {
12     // બટિ 4 સેટ છે
13 }

```

### મેમરી ટ્રીક

"AND ક્લિયર કરે, OR સેટ કરે, XOR ટોગલ કરે, શિફ્ટ ગુણાકાર/ભાગાકાર કરે"

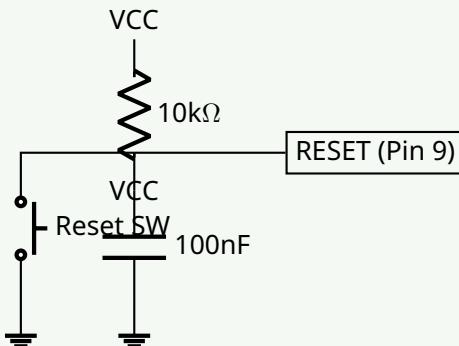
## પ્રશ્ન 3(a OR) [3 ગુણ]

ATmega32 માઇકોકન્ટ્રોલર માટે રીસેટ સર્કિટ સમજાવો.

### જવાબ

રીસેટ સર્કિટ પાવર લાગુ થાય ત્યારે અથવા સિસ્ટમ રીસેટ દરમિયાન ATmega32નું ચોંગ ઇનિશિયલાઇઝેશન સુનિશ્ચિત કરે છે.

આફ્ટિ 6. Reset Circuit



- એક્ટિવ-લો RESET: માઇકોકન્ટ્રોલરને રીસેટ કરવા માટે લો રાખવું જોઈએ
- એક્સ્ટરનલ રીસેટ: મેન્યુઅલ રીસેટ બટન RESET પિનને ગ્રાઉન્ડ સાથે જોડે છે
- પાવર-ઓન રીસેટ: પાવર પ્રથમ વખત લાગુ થાય ત્યારે ઓનો-રીસેટ
- બ્રાઉન-આઉટ ડિટેક્શન: વોલ્ટેજ થ્રેશોફ્ટથી નીચે જાય ત્યારે રીસેટ
- વોચડોગ ટાઇમર: સોફ્ટવેર મલિન્કશન પર રીસેટ

### મેમરી ટ્રીક

“પુલ અપ, પુશ બટન, પાવર સ્ટાર્ટ, વોલ્ટેજ ફોષ્પ”

## પ્રશ્ન 3(b OR) [4 ગુણ]

EEPROM સાથે સંકળાયેલ રજીસ્ટરોની યાદી બનાવો અને ATmega32 ના EEPROM ને ઈન્ટરફેસ કરવા માટે પગલાંઓ લખો.

### જવાબ

ATmega32માં ઓન-ચિપ EEPROM છે જેના એક્સેસ કંટ્રોલ માટે ડેડિકેટ રજિસ્ટર્સ છે.

કોષ્ટક 12. EEPROM રજિસ્ટર્સ

| રજિસ્ટર     | ફુલ નામ                 |
|-------------|-------------------------|
| EEARH/EEARL | EEPROM એડ્રેસ રજિસ્ટર્સ |
| EEDR        | EEPROM ડેટા રજિસ્ટર     |
| EECR        | EEPROM કંટ્રોલ રજિસ્ટર  |

EEPROM ઈન્ટરફેસ કરવાના પગલાં:

- પૂર્ણતા માટે રાહ જુઓ:
  - ચેક કરો કે અગાઉની રાઇટ ઓપરેશન પૂર્ણ થઈ છે કે નહીં (EECR માં EWE બિટ)
- એડ્રેસ સેટ કરો:
  - EEARH:EEARL માં એડ્રેસ લોડ કરો (16-બિટ એડ્રેસ)
- રીડ અથવા રાઇટ ઓપરેશન:
  - રીડ માટે: EERE માં EERE બિટ સેટ કરો, પછી EEDR વાંચો
  - રાઇટ માટે: EEDR માં ડેટા લખો, પછી EECR માં EEMWE અને EWE બિટ્સ સેટ કરો
- પૂર્ણતા માટે રાહ જુઓ:
  - EEWE બિટ જીરો થાય ત્યાં સુધી પોલ કરો

## મેમરી ટ્રીક

“રાહ જુઓ, એડ્રેસ, ડેટા, કંટ્રોલ, રાહ જુઓ”

## પ્રશ્ન 3(c OR) [7 ગુણ]

PORTC.2 પિન પર 1KHz ની સ્કવેર વેવ જનરેટ કરવા માટે C પ્રોગ્રામ લખો. delay બનાવવા માટે Timer0, Normal mode અને 1:8 પ્રી-સ્કેલરનો ઉપયોગ કરો. CRYSTAL FREQ. = 8 MHz ધરો.

## જવાબ

## જવાબ:

## Code

```

1 #include <avr/io.h>
2
3 int main(void)
4 {
5     // PORTC.2 ને આઉટપુટ તરીકે કન્ફાર્મિંગ કરો
6     DDRC |= (1 << 2); // PC2 ને આઉટપુટ તરીકે સેટ કરો
7
8     // Timer0 કન્ફાર્મિંગ - નોર્મલ મોડ, 1:8 પરિસ્કેલર
9     TCCR0 = (0 << WGM01) | (0 << WGM00) | (0 << CS02) | (1 << CS01) | (0 << CS00);
10
11    // 1KHz માટે ટાઇમર વેલ્યુની ગણતરી (500s પીરથિડ, 250s હાફ્પીરથિડ)
12    // 8MHz/8 = 1MHz ટાઇમર કલોક, 250 સાઈકલ્સ ફીર 250s
13    // 256-250 = 6 (250s માટે સ્ટાર્ટિંગ વેલ્યુ)
14
15    while (1)
16    {
17        // PORTC.2 ટોગલ કરો
18        PORTC ^= (1 << 2);
19
20        // ટાઇમર રીસેટ કરો
21        TCNT0 = 6;
22
23        // ટાઇમર ઓવરફ્લો થાય ત્યાં સુધી રાહ જુઓ
24        while (!(TIFR & (1 << TOV0)));
25
26        // ઓવરફ્લો ફૂલેગ ફૂલથિર કરો
27        TIFR |= (1 << TOV0);
28    }
29
30    return 0;
31 }
```

- ફિક્વન્સી ગણતરી: 1KHz = 1000Hz = 1ms પીરથિડ =  $500\mu s$  હાફ-પીરથિડ
- ટાઇમર કલોક:  $8MHz \div 8 = 1MHz = 1\mu s$  પ્રતિ ટિક
- ટાઇમર ટિક્સ:  $250\mu s \div 1\mu s = 250$  ટિક્સ
- ઇનિશિયલ વેલ્યુ:  $256 - 250 = 6$  (250 ટિક્સ પછી ઓવરફ્લો માટે)

## મેમરી ટ્રીક

“કન્ફિગર, કેલ્ક્યુલેટ, ટોગલ, રીસેટ, વેઇટ, કિલ્યર, રિપીટ”

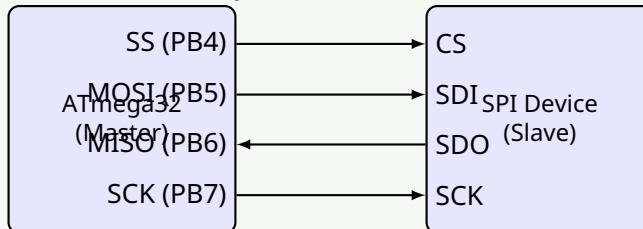
## પ્રશ્ન 4(a) [3 ગુણ]

ATmega32 સાથે SPI આધારિત device ઇન્ટરફેસિંગ ડાયાગ્રામ દોરો અને સમજાવો.

### જવાબ

SPI (સીરિયલ પેરિફરલ ઇન્ટરફેસ) એ સિંકોન્સ સીરિયલ કમ્યુનિકેશન પ્રોટોકોલ છે જે ATmega32ને પેરિફરલ ડિવાઇસ સાથે ઇન્ટરફેસ કરવા માટે વપરાય છે.

આકૃતિ 7. SPI Interfacing



- MOSI (માસ્ટર આઉટ સ્લેવ ઇન): માસ્ટરથી સ્લેવ સુધી ડેટા
- MISO (માસ્ટર ઇન સ્લેવ આઉટ): સ્લેવથી માસ્ટર સુધી ડેટા
- SCK (સીરિયલ કલોક): માસ્ટર દ્વારા પ્રદાન કરેલ સિંકનાઇઝેશન કલોક
- SS (સ્લેવ સિલેક્ટ): ચોક્કસ સ્લેવ ડિવાઇસ પસંદ કરવા માટે એક્ટિવ-લો સિન્ઘલ

### મેમરી ટ્રીક

“માસ્ટર આઉટપુટ્સ, સ્લેવ ઇનપુટ્સ, કલોક કીન્સ સિંકનાઇઝેશન”

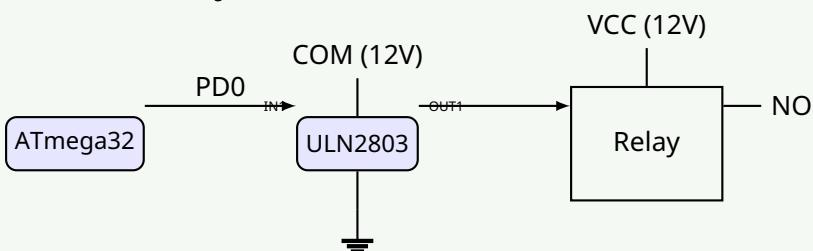
### પ્રશ્ન 4(b) [4 ગુણ]

ATmega32 સાથે ULN2803 નો ઉપયોગ કરીને રિલેનું ઇન્ટરફેસિંગ દોરો અને સમજાવો.

### જવાબ

ULN2803 એ ડાલિંગટન ટ્રાન્ઝિસ્ટર પેર્સનો એરે છે જે માઇકોકન્ટ્રોલર પિન્સથી રિલે જેવા હાઇ-કરંટ ડિવાઇસને ડ્રાઇવ કરવા માટે વપરાય છે.

આકૃતિ 8. Relay Interfacing using ULN2803



- કરંટ એમ્પિલફિકેશન: ULN2803 પ્રતિ ચેનલ 500mA સુધી સિંક કરી શકે છે
- વોલ્ટેજ આઇસોલેશન: બિલ્ટ-ઇન ડાયોડ્સ ઇન્ડિકેટ કિકબેક સામે સુરક્ષા આપે છે
- મણિટ્યુલ ચેનલ્સ: એક પેકેજમાં 8 ડાલિંગટન પેર્સ
- હાઇ વોલ્ટેજ રેટિંગ: આઉટપુટ પર 50V સુધી હેન્ડલ કરી શકે છે

### મેમરી ટ્રીક

“લો કરંટ કંટ્રોલ્સ હાઇ કરંટ લોડ્સ”

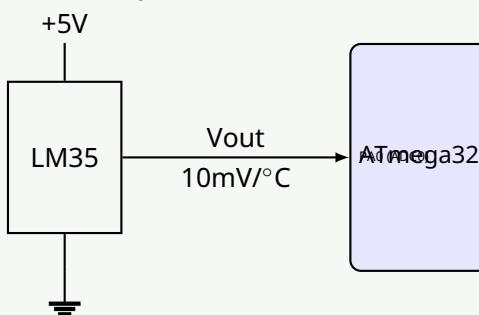
### પ્રશ્ન 4(c) [7 ગુણ]

ATmega32 ના ADC0 (પિન 40) પર જોડાયેલ LM35 નો ઇન્ટરફેસિંગ ડાયાગ્રામ દોરો અને PORT-B પર ADC નું ડિજિટલ પરિણામ દર્શાવવા માટે AVR C પ્રોગ્રામ લખો. (8-બીટ મોડમાં ADC નો ઉપયોગ કરો).

## જવાબ

LM35 એ પ્રેસિજન તાપમાન સેન્સર છે જે તાપમાનના પ્રમાણમાં એનાલોગ વોલ્ટેજ આઉટપુટ આપે છે.

આકૃતિ 9. LM35 Interfacing



C પ્રોગ્રામ:

### Code

```

1 #include <avr/io.h>
2 #include <util/delay.h>
3
4 int main(void)
{
5     // PORTB ને પરિષામ દર્શાવવા માટે આઉટપુટ તરીકે કન્ફિર્મિર કરો
6     DDRB = 0xFF;
7
8     // ADC કન્ફિર્મિર કરો
9     ADMUX = (0 << REFS1) | (1 << REFS0) | // AVCC as રેફરન્સ
10    (1 << ADLAR) | // બટિસ્ટ- માટે લેફ્ટ એડજસ્ટ રાઇલ્સ
11    (0 << MUX4) | (0 << MUX3) | (0 << MUX2) | (0 << MUX1) | (0 << MUX0); // ADC0
12
13    ADCSRA = (1 << ADEN) | // ADC એનેબલ કરો
14    (1 << ADPS2) | (1 << ADPS1) | (1 << ADPS0); // પ્રીસ્કૉલર 128
15
16    while (1)
17    {
18        // કન્ફર્મન શરૂ કરો
19        ADCSRA |= (1 << ADSC);
20
21        // કન્ફર્મન પૂરાણ થાય ત્યાં સુધી રાહ જુઓ
22        while (ADCSRA & (1 << ADSC));
23
24        // PORTB પર પરિષામ દર્શાવો (ADCH માંથી બટિસ્ટ-
25        PORTB = ADCH;
26
27        // અણલા રીડિંગ પહેલા રાહ જુઓ
28        _delay_ms(500);
29    }
30
31    return 0;
32}
33

```

- તાપમાન ગણતરી: LM35 10mV/°C આઉટપુટ આપે છે
- ADC કન્ફિગ્રેશન: 8-બિટ રીડિંગ માટે લેફ્ટ-એડજસ્ટેડ
- રેઝાલ્યુશન: 5V રેફરન્સ સાથે 8-બિટ મોડનો ઉપયોગ કરવાથી આશરે 1°C રેઝાલ્યુશન મળે છે
- રેન્જ: 0-255°C રેન્જ માપી શકે છે (8-બિટ રાન્જિસ્ટર દ્વારા મર્યાદિત)

### મેમરી ટ્રીક

"કનેક્ટ, કન્ફિગર, કન્વર્ટ, કેપ્ચર, ડિસ્પ્લે"

## પ્રશ્ન 4(a OR) [3 ગુણ]

PORTA ના PA0 પિનને સતત મોનિટર કરવા માટે AVR C પ્રોગ્રામ લખો. જો તે HIGH હોય, તો PORTC ના PC0 પિન પર HIGH મોકલો; નહિંતર, PORTC ના PC0 પિન પર LOW મોકલો.

### જવાબ

#### જવાબ:

##### Code

```

1 #include <avr/io.h>
2
3 int main(void)
4 {
5     // PA0 ને ઇનપુટ તરીકે કન્ફિગિર કરો
6     DDRA &= ~(1 << PA0);
7
8     // PA0 પર પુલઅપ-રેજિસ્ટર એનેબલ કરો
9     PORTA |= (1 << PA0);
10
11    // PC0 ને આઉટપુટ તરીકે કન્ફિગિર કરો
12    DDRC |= (1 << PC0);
13
14    while (1)
15    {
16        // ચેક કરો કે PA0 HIGH છે કે નહીં
17        if (PINA & (1 << PA0))
18        {
19            // PC0 ને HIGH સેટ કરો
20            PORTC |= (1 << PC0);
21        }
22        else
23        {
24            // PC0 ને LOW સેટ કરો
25            PORTC &= ~(1 << PC0);
26        }
27    }
28
29    return 0;
30 }
```

- ઇનપુટ કન્ફિગરેશન: પુલ-અપ રેજિસ્ટર સાથે ઇનપુટ તરીકે સેટ કરો
- કન્ફિગ્યુર મોનિટરિંગ: ઇન્ફિનિટ લૂપ પિન સ્ટેટ ચેક કરે છે
- આઉટપુટ એક્ઝિટ્યુન્ટ: PC0 PA0 સ્ટેટનું મિરરિંગ કરે છે
- ઇન્ફિશિયન્ટ કોડ: પિન મોનિટરિંગ માટે સિમ્પલ કન્ડિશનલ સ્ટેટમેન્ટ

### મેમરી ટ્રીક

“કન્ફિગાર, મોનિટર, મિરર”

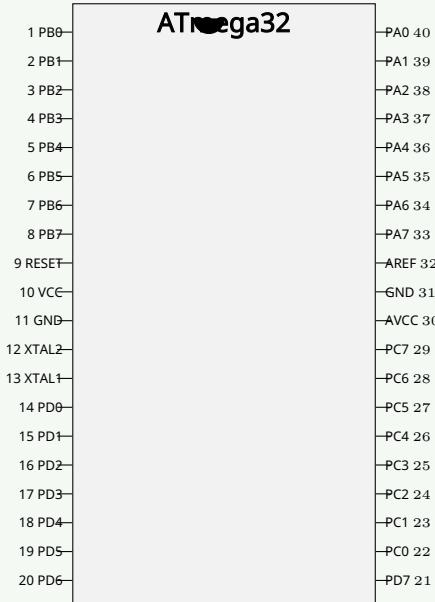
## પ્રશ્ન 4(b OR) [4 ગુણ]

ATmega32 પિન ડાયાગ્રામ દોરો અને Vcc, AVcc અને Aref પિનનાં કાર્ય લખો.

### જવાબ

ATmega32માં 40 પિન્સ DIP પેકેજમાં ગોઠવાયેલ છે, જેમાં પાવર સપ્લાય પિન્સ અલગ-અલગ ફુંક્શન ધરાવે છે.

આફ્રતિ 10. ATmega32 Pin Diagram



કોષ્ટક 13. પાવર સપ્લાય પિન્સ

| પિન  | ફુંક્શન        | વર્ણન  |
|------|----------------|--|
| VCC  | ડિજિટલ પાવર    | ડિજિટલ સર્કિટ્સ માટે મુખ્ય સપ્લાય વોલ્ટેજ (5V ટિપિકલ)    |
| AVCC | એનાલોગ પાવર    | એનાલોગ સર્કિટ્ટરી માટે સપ્લાય, ખાસ કરીને ADC (5V ટિપિકલ) |
| AREF | એનાલોગ રેફરન્સ | ADC માટે એક્સટર્નલ રેફરન્સ વોલ્ટેજ                       |

- VCC: ડિજિટલ લોજિક અને I/O પોર્ટ્સને પાવર આપે છે
- AVCC: ADC બિન-વપરાશમાં હોય તો પણ, VCC ની 0.3V ની અંદર હોવું જોઈએ
- AREF: ADC માટે વૈકલ્પિક એક્સટર્નલ રેફરન્સ, અન્યથા AVCC સાથે કનેક્ટ કરો

#### મેમરી ટ્રીક

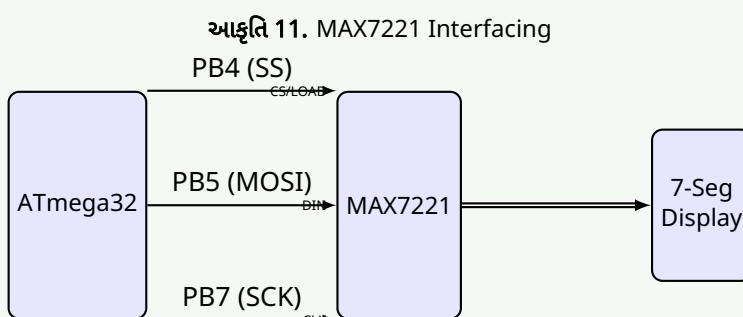
"VCC કોર સર્કિટ્સ માટે, AVCC એનાલોગ માટે, AREF રેફરન્સ માટે"

## પ્રશ્ન 4(c OR) [7 ગુણ]

ATmega32 સાથે MAX7221 નું ઇન્ટરફેસિંગ દોરો અને સમજાવો.

#### જવાબ

MAX7221 એ LED ડિસ્પ્લે ફ્રાઇવર IC છે જે SPI કમ્યુનિકેશનનો ઉપયોગ કરીને ATmega32 સાથે ઇન્ટરફેસ કરે છે.



કોષ્ટક 14. કનેક્શન વિગતો

| ATmega32 પિન | MAX7221 પિન | ફંક્શન                           |
|--------------|-------------|----------------------------------|
| PB4 (SS)     | CS/LOAD     | ચિપ સિલેક્ટ/લોડ ડેટા             |
| PB5 (MOSI)   | DIN         | MAX7221માં ડેટા ઇનપુટ            |
| PB6 (MISO)   | DOUT        | ડેટા આઉટપુટ (ધારીવાર બિનાઉપયોગી) |
| PB7 (SCK)    | CLK         | કલોક સિગનલ                       |

### ઇન્ટરફેસિંગ સ્ટેપ્સ:

1. SPI ઇનિશિયલાઇઝ કરો:
  - SPI ને માસ્ટર મોડમાં કન્ફિગાર કરો
  - ચોગ્ય કલોક પોલેરિટી અને ફેઝ સેટ કરો
  - SS (PB4) ને આઉટપુટ તરીકે અને પ્રારંભિક રીતે હાઇ સેટ કરો
2. MAX7221 ઇનિશિયલાઇઝ કરો:
  - ડિકોડ મોડ સેટ કરો (BCD ડિકોડ અથવા નો-ડિકોડ)
  - સ્કેન લિમિટ (ડિજિટ્સની સંખ્યા) સેટ કરો
  - ઇન્ટેન્સિટી (બાઇટનેસ) સેટ કરો
  - ડિસ્પલે ચાલુ કરો
3. ડેટા મોકલો:
  - SS ને લો પૂલ કરો
  - રજિસ્ટર એન્ડ્રેસ પછી ડેટા મોકલો
  - ડેટા લેંચ કરવા માટે SS ને હાઇ પૂલ કરો

### Code

```

1 // Initialization code example
2 void MAX7221_init() {
3     // Initialize SPI
4     DDRB |= (1<<PB4)|(1<<PB5)|(1<<PB7); // Set SS, MOSI, SCK as outputs
5     SPCR = (1<<SPE)|(1<<MSTR)|(1<<SPR0); // Enable SPI, Master, clk/16
6
7     // Initialize MAX7221
8     MAX7221_send(0x09, 0xFF); // Decode mode: BCD for all digits
9     MAX7221_send(0x0A, 0x0F); // Intensity: 15/32 Duty (Max)
10    MAX7221_send(0x0B, 0x07); // Scan limit: Display all digits
11    MAX7221_send(0x0C, 0x01); // Shutdown mode: Normal Operation
12    MAX7221_send(0x0F, 0x00); // Display test: Normal Operation
13 }

```

### મેમરી ટ્રીક

"સેન્સ, સિલેક્ટ, કલોક, ડેટા, ડિસ્પલે"

### પ્રશ્ન 5(a) [3 ગુણ]

L293D મોટર દ્રાઇવર IC નો પિન ડાયાગ્રામ દોરો અને સમજાવો.

### જવાબ

L293D એ DC મોટર્સના બાયડાયરેક્શનલ કંટ્રોલ માટે ડિઝાઇન કરાયેલ કવાડુપલ હાફ-H દ્રાઇવર છે.

આકૃતિ 12. L293D Pin Diagram



કોષ્ટક 15. L293D પિન ફંક્શન-સ

| પિન          | નામ       | ફંક્શન                               |
|--------------|-----------|--------------------------------------|
| 1, 9         | EN1, EN2  | એનેબલ ઇનપુટ્સ (PWM સિગલ હોઈ શકે છે)  |
| 2, 7, 10, 15 | IN1-IN4   | લોજિક ઇનપુટ્સ                        |
| 3, 6, 11, 14 | OUT1-OUT4 | મોટર્સ કનેક્ટ કરવા માટે આઉટપુટ પિન્સ |
| 4, 5, 12, 13 | GND       | ગ્રાઉન્ડ કનેક્શન-સ                   |
| 8            | VCC2      | મોટર સપ્લાય વોલ્ટેજ (4.5V-36V)       |
| 16           | VCC1      | લોજિક સપ્લાય વોલ્ટેજ (5V)            |

- ઝુઅલ H-બિજ: બે DC મોટર્સને સ્વતંત્ર રીતે કંટ્રોલ કરી શકે છે
- હીટ સિક: ગ્રાઉન્ડ પિન્સ હીટ ડિસિપેશન પ્રદાન કરે છે
- હાઇ કરેટ: પ્રતિ ચેનલ 600mA સુધી ડ્રાઇવ કરી શકે છે
- પ્રોટેક્શન ડાયોડ્સ: ઇન્ડક્ટિવ લોડ્સ માટે ઇન્ટરનલ ફ્લાયબેક ડાયોડ્સ

### મેમરી ટ્રીક

"એનેબલ, ઇનપુટ, આઉટપુટ, પાવર"

## પ્રશ્ન 5(b) [4 ગુણ]

ADMUX રજિસ્ટર દોરો અને સમજાવો.

### જવાબ

ADMUX (ADC માલ્ટિપ્લિક્સર સિલેક્શન રજિસ્ટર) ATmega32માં એનાલોગ ચેનલ સિલેક્શન અને રિજાલ્ટ ફોર્મોટ કંટ્રોલ કરે છે.

આકૃતિ 13. ADMUX Register

|            |            |            |         |           |           |           |           |
|------------|------------|------------|---------|-----------|-----------|-----------|-----------|
| REFS1<br>7 | REFS0<br>6 | ADLAR<br>5 | --<br>4 | MUX3<br>3 | MUX2<br>2 | MUX1<br>1 | MUX0<br>0 |
|------------|------------|------------|---------|-----------|-----------|-----------|-----------|

કોષ્ટક 16. ADMUX બિટ ફંક્શન-સ

| બિટ્સ | નામ     | ફંક્શન                   |
|-------|---------|--------------------------|
| 7:6   | REFS1:0 | રેફરન્સ વોલ્ટેજ સિલેક્શન |
| 5     | ADLAR   | ADC લેફ્ટ એડજસ્ટ રિજાલ્ટ |
| 3:0   | MUX3:0  | એનાલોગ ચેનલ સિલેક્શન     |

### REFS1:0 સેટિંગ્સ:

- 00: AREF પિન (એક્સ્ટર્નલ રેફરન્સ)
- 01: એક્સ્ટર્નલ કેપેસિટર સાથે AVCC

- 11: ઇન્ટરનલ 2.56V રેફરન્સ
- ચેનલ સિલેક્શન: MUX3:0 ક્રાન ADC ઇનપુટને કનેક્ટ કરવું તે સિલેક્ટ કરે છે
- રિજાલ્ટ એલાઇનમેન્ટ: ADLAR=1 રિજાલ્ટને લેફ્ટ શિફ્ટ કરે છે (8-બિટ રીડિંગ્સ માટે)
- ડિફરેન્શિયલ ઇનપુટ્સ: કેટલાક MUX કોમ્પિનેશન્સ ડિફરેન્શિયલ મેઝરમેન્ટ્સની મંજૂરી આપે છે

### મેમરી ટ્રીક

"રેફરન્સ, એલાઇનમેન્ટ, મલ્ટિપ્લેક્સર"

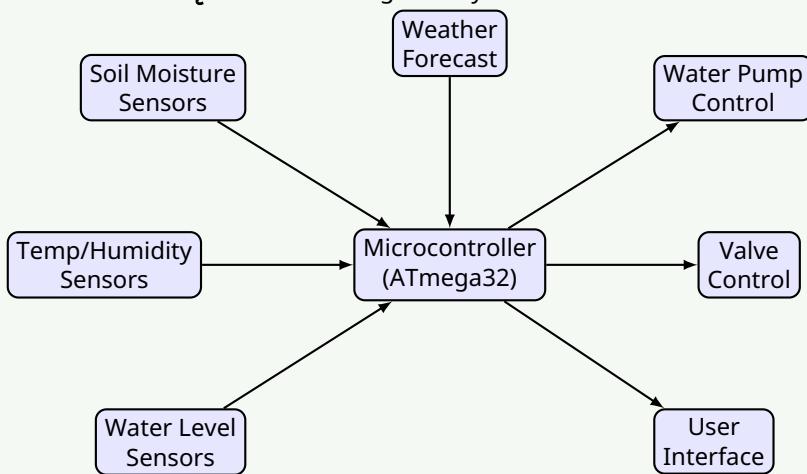
## પ્રશ્ન 5(c) [7 ગુણ]

સ્માર્ટ સિંચાઈ પદ્ધતિ સમજાવો.

### જવાબ

સ્માર્ટ સિંચાઈ સિસ્ટમ પર્યાવરણીય પરિસ્થિતિઓના આધારે વનસ્પતિ ખેતી માટે પાણીનું કાર્યક્ષમ રીતે વ્યવસ્થાપન કરવા એંડ્રોઇડ ટેકનોલોજીનો ઉપયોગ કરે છે.

આકૃતિ 14. Smart Irrigation System Flowchart



કોષ્ટક 17. સ્માર્ટ સિંચાઈ કોમ્પોનેન્ટ્સ

| કોમ્પોનેન્ટ            | ફુંક્શન  |
|------------------------|--|
| સોઇલ મોઇશ્યુર સેન્સર્સ | જમીનમાં પાણીનું પ્રમાણ માપે છે                       |
| તાપમાન/ભેજ સેન્સર્સ    | પર્યાવરણીય પરિસ્થિતિઓનું મોનિટરિંગ કરે છે            |
| વાહ્યસ                 | અલગ અલગ જોન માટે વોટર ફ્લો કંટ્રોલ કરે છે            |
| પાય કંટ્રોલ            | જરૂર પડે ત્યારે વોટર પાય એક્ટિવેટ કરે છે             |
| માઇકોકન્ટ્રોલર         | સેન્સર ડેટા પ્રોસેસ કરે છે અને આઉટપુટ કંટ્રોલ કરે છે |
| યુગર ઇન્ટરફેસ          | મોનિટરિંગ અને મેન્યુઅલ કંટ્રોલની મંજૂરી આપે છે       |

### કી ફીચર્સ:

1. ઓટોમેટેડ વોટરિંગ: જ્યારે સોઇલ મોઇશ્યુર થ્રેશોલ્ડથી નીચે જાય ત્યારે જ વનસ્પતિઓને પાણી આપે છે
2. વેધર એડાપ્ટેશન: તાપમાન, ભેજ અને વરસાદ ફોરકાસ્ટના આધારે વોટરિંગ શેડ્યુલ એડજસ્ટ કરે છે
3. જોન કંટ્રોલ: અલગ અલગ વિસ્તારોમાં અલગ અલગ વોટરિંગ શેડ્યુલ હોઈ શકે છે
4. વોટર કન્ઝર્વેશન: ઓપ્ટિમલ પ્લાન ગ્રોથ માટે મિનિમન જરૂરી પાણીનો ઉપયોગ કરે છે
5. રિપોટ મોનિટરિંગ: સિસ્ટમ ર્ટેટસ અને કંટ્રોલ માટે મોબાઇલ એપ અથવા વેબ ઇન્ટરફેસ
6. શેડ્યુલિંગ: ટાઇમ-વેજા અને કન્ડિશન-વેજા વોટરિંગ ઓપ્શન્સ

### મેમરી ટ્રીક

"સેન્સ, ડિસાઇટ, કન્ઝર્વ, ગ્રો"

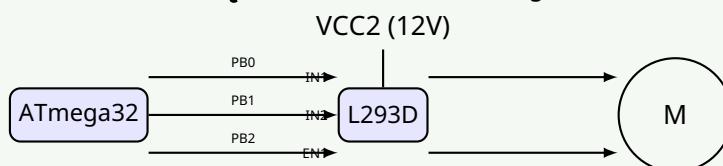
## પ્રશ્ન 5(a OR) [3 ગુણ]

L293D મોટર દ્વારા ઉપયોગ કરીને ATmega32 સાથે DC મોટરને ઇન્ટરફેસ કરવા માટે સર્કિટ ડાયાગ્રામ દોરો.

### જવાબ

સર્કિટ DC મોટરને બાયડાયરેક્શનલ કંટ્રોલ માટે L293D મારફત ATmega32 સાથે કનેક્ટ કરે છે.

આકૃતિ 15. DC Motor Interfacing



### કંટ્રોલ લોજિક:

કોષ્ટક 18. મોટર કંટ્રોલ લોજિક

| IN1 | IN2 | EN1 | મોટર સ્ટેટ્સ            |
|-----|-----|-----|-------------------------|
| 0   | 0   | 1   | સ્ટોપ (બ્રેક)           |
| 1   | 0   | 1   | કલોકવાઈજ રોટેશન         |
| 0   | 1   | 1   | કાઉન્ટર-કલોકવાઈજ રોટેશન |
| 1   | 1   | 1   | સ્ટોપ (બ્રેક)           |
| X   | X   | 0   | મોટર ડિસેબલ્ડ           |

- સ્પીડ કંટ્રોલ: EN1 પર PWM સિગનલ મોટરની સ્પીડ કંટ્રોલ કરી શકે છે
- ડિરેક્શન કંટ્રોલ: IN1 અને IN2 રોટેશન ડિરેક્શન કંટ્રોલ કરે છે
- પાવર સેપરેશન: લોજિક માઇકોકન્ટ્રોલર દ્વારા, મોટર અલગ સપ્લાય દ્વારા પાવર્ડ

### મેમરી ટ્રીક

"એનેબલ અને ડિરેક્શન કંટ્રોલ મોટર"

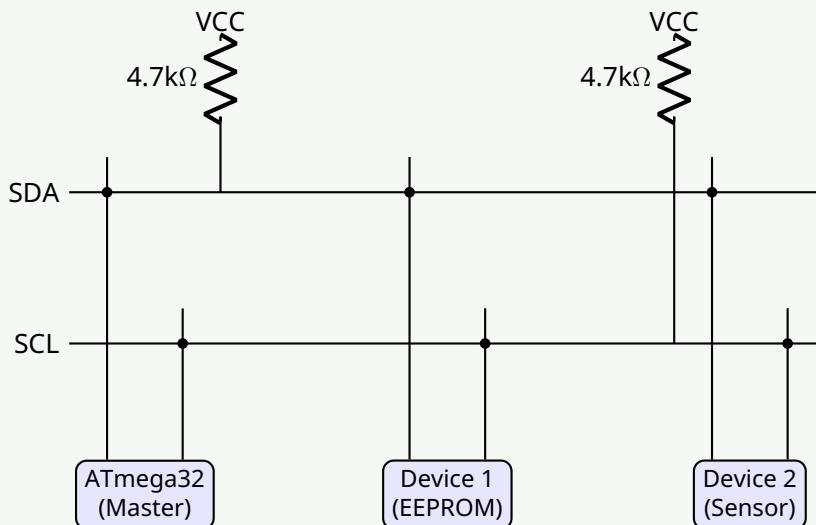
## પ્રશ્ન 5(b OR) [4 ગુણ]

ATmega32 સાથે I2C આધારિત device ઇન્ટરફેસિંગ ડાયાગ્રામ દોરો અને સમજાવો.

### જવાબ

I2C (ઇન્ટર-ઇન્ટરગ્રેટેડ સર્કિટ) એ માઇકોકન્ટ્રોલર સાથે મલ્ટિપલ ડિવાઇસ કનેક્ટ કરવા માટે ટુ-વાયર સીરિયલ બસ છે.

આકૃતિ 16. I2C Interfacing



#### કી કોમ્પોનન્ટ્સ:

- SDA (સીરિયલ ડેટા લાઇન): બાયડાયરેક્શનલ ડેટા ટ્રાન્સફર લાઇન
- SCL (સીરિયલ કલોક લાઇન): માસ્ટર દ્વારા જનરેટ કરેલ કલોક સિગ્નલ
- પુલ-અપ રેજિસ્ટર્સ: બંને લાઇન્સ પર જરૂરી (સામાન્ય રીતે  $4.7\text{k}\Omega$ )
- મલ્ટિપલ ડિવાઇસીસ: દરેક I2C ડિવાઇસ યુનિક એડ્રેસ ધરાવે છે

#### કમ્પ્યુનિકેશન પ્રોસેસ:

1. સ્ટાર્ટ કન્ડિશન: SCL હાઇ હોય ત્યારે SDA હાઇ-ટુ-લો ટ્રાન્ઝિશન કરે છે
2. એડ્રેસ ટ્રાન્સમિશન: 7-બિટ ડિવાઇસ એડ્રેસ પછી R/W બિટ
3. એકનોલેજમેન્ટ: રિસીવિંગ ડિવાઇસ SDA ને પુલ ડાઉન કરે છે
4. ડેટા ટ્રાન્સફર: એકનોલેજમેન્ટ સાથે 8-બિટ ડેટા બાઇટ્સ
5. સ્ટોપ કન્ડિશન: SCL હાઇ હોય ત્યારે SDA લો-ટુ-હાઇ ટ્રાન્ઝિશન કરે છે

#### મેમરી ટ્રીક

“સ્ટાર્ટ, એડ્રેસ, એકનોલેજ, ડેટા, સ્ટોપ”

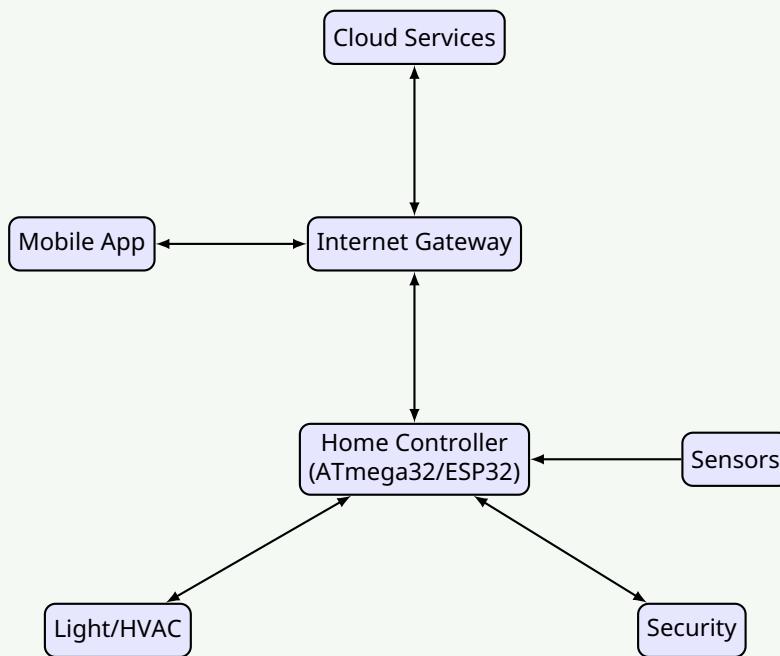
## પ્રશ્ન 5(c OR) [7 ગુણ]

IOT આધારિત હોમ ઓટોમેશન સિસ્ટમ સમજાવો.

#### જવાબ

IOT-આધારિત હોમ ઓટોમેશન સિસ્ટમ ઘરના ઉપકરણોને રિમોટ મોનિટરિંગ અને કંટ્રોલ માટે ઇન્ટરનેટ સાથે કનેક્ટ કરે છે.

આકૃતિ 17. IoT Home Automation Architecture



કોષ્ટક 19. હોમ ઓટોમેશન કોમ્પોનેન્ટ્સ

| કોમ્પોનેન્ટ      | ફંક્શન   |
|------------------|--|
| કન્ટ્રોલર        | સેન્ટ્રલ પ્રોસેસિંગ યુનિટ (માઇક્રોકન્ટ્રોલર/SBC) |
| સેન્સર્સ         | તાપમાન, મોશન, લાઇટ, બેજનું મોનિટરિંગ કરે છે      |
| એક્સચ્યુયાન્ડર્સ | લાઇટ્સ, ઉપકરણો, લોક્સ, HVAC કન્ટ્રોલ કરે છે      |
| ગેટવે            | ઇન્ટરનેટ અને લોકલ ડિવાઇસ સાથે કનેક્ટ થાય છે      |
| યુઝર ઇન્ટરફેસ    | મોબાઇલ એપ, વોઇસ કન્ટ્રોલ, વેબ ડેશબોર્ડ           |
| કલાઉડ સર્વિસીસ   | ડાટા સ્ટોરેજ, પ્રોસેસિંગ અને રિમોટ એક્સેસ        |

## કી ફીચર્સ:

- રિમોટ એક્સેસ: ગમે ત્યાંથી ઘરના ઉપકરણો કન્ટ્રોલ કરવા
- વોઇસ કન્ટ્રોલ: વોઇસ આસિસ્ટન્ટ્સ (એલેક્સા, ગૂગલ હોમ) સાથે ઇન્ટિગ્રેશન
- એનજ્ઞ મેનેજમેન્ટ: પાવર કન્જમ્પશનનું મોનિટરિંગ અને ઓપ્ટિમાઇઝેશન
- સિક્યુરિટી: દરવાજા, બારી અને કેમેરાનું કન્ટ્રોલ અને મોનિટરિંગ
- શરૂઆતિંગ: સમય અથવા ઇવેન્ટ્સના આધારે ડિવાઇસના ઓપરેશનનું ઓટોમેશન
- સીન સેટિંગ: મલ્ટિપલ ડિવાઇસ માટે પ્રીડિફાઇન્ડ કન્ફિગરેશન
- એડિટિવ કન્ટ્રોલ: યુઝર પ્રેફરન્સીસ અને પેર્ટન શીખવાનું અને અનુકૂલન કરવાનું

## મેમરી ટ્રીક

“કનેક્ટ, કન્ટ્રોલ, મોનિટર, ઓટોમેટ, લર્ન”