

Subject Name (Gujarati)

4343204 -- Summer 2025

Semester 1 Study Material

Detailed Solutions and Explanations

પ્રશ્ન 1(અ) [3 ગુણ]

રીઅલ ટાઇમ ઓપરેટિંગ સિસ્ટમની લાક્ષણિકતાઓની ચર્ચા કરો.

જવાબ

Table 1: RTOS લાક્ષણિકતાઓ

લાક્ષણિકતા	વર્ણન
નિર્ધારિત વર્તન	અનુમાનિત પ્રતિસાદ સમય
સમય મર્યાદા	કઠિન અને નરમ ડેડલાઇન
પ્રાથમિકતા શેડ્યુલિંગ	પ્રાથમિકતા દ્વારા કાર્ય અમલ
સંસાધન વ્યવસ્થાપન	કાર્યક્ષમ મેમરી અને CPU ઉપયોગ

- નિર્ધારિત વર્તન: સિસ્ટમ ગેરંટીવાળા સમય મર્યાદામાં પ્રતિસાદ આપે છે
- મલ્ટિટાસ્કિંગ સપોર્ટ: બહુવિધ કાર્યો પ્રાથમિકતા સાથે સમાંતર ચાલે છે
- ઇન્ટરપ્ટ હેન્ડલિંગ: બાહ્ય ઘટનાઓને ઝડપી પ્રતિસાદ

સ્મરણ સહાયક: "RTOS કાર્યો યોગ્ય રીતે વિતરિત કરે છે"

પ્રશ્ન 1(બ) [4 ગુણ]

AVR I/O પોર્ટ રજિસ્ટરનું વર્ણન કરો.

જવાબ

Table 2: AVR I/O પોર્ટ રજિસ્ટર

રજિસ્ટર	કાર્ય	પ્રવેશ
DDRx	ડેટા દિશા રજિસ્ટર	વાંચો/લખો
PORTx	પોર્ટ આઉટપુટ રજિસ્ટર	વાંચો/લખો
PINx	પોર્ટ ઇનપુટ રજિસ્ટર	ફક્ત વાંચો

- DDRx રજિસ્ટર: પિન દિશા નિયંત્રિત કરે છે (0=ઇનપુટ, 1=આઉટપુટ)
- PORTx રજિસ્ટર: આઉટપુટ મૂલ્યો સેટ કરે છે અથવા pull-up રેઝિસ્ટર સક્રિય કરે છે
- PINx રજિસ્ટર: ઇનપુટ ઓપરેશન માટે વર્તમાન પિન સ્થિતિ વાંચે છે

સ્મરણ સહાયક: "દિશા, પોર્ટ, પિન - DPP"

પ્રશ્ન 1(ક) [7 ગુણ]

વિવિધ AVR માઇક્રોકન્ટ્રોલરની સરખામણી કરો અને એમ્બેડેડ સિસ્ટમ માટે માઇક્રોકન્ટ્રોલર પસંદ કરવા માટે કયા પરિબલો ધ્યાનમાં લેવા જોઈએ?

જવાબ

Table 3: AVR માઇક્રોકન્ટ્રોલર સરખામણી

જાદીયા

Table 4: SRAM વિ EEPROM સરખામણી

પેરામીટર	SRAM	EEPROM
કદ	2KB	1KB
અસ્થિરતા	અસ્થિર	બિન-અસ્થિર
પ્રવેશ ઝડપ	ઝડપી	ધીમી
લેખન ચક્ર	અમર્યાદિત	100,000 ચક્ર

- ડેટા રીટેન્શન: SRAM પાવર-ઓફ પર ડેટા ખોવાય છે, EEPROM ડેટા જાળવે છે
 - ઉપયોગ હેતુ: SRAM વેરિએબલ માટે, EEPROM કોન્ફિગરેશન ડેટા માટે
- સ્મરણ સહાયક:** "SRAM ઝડપી પણ ભૂલી જાય, EEPROM ટકી રહે"

પ્રશ્ન 2(બ) [4 ગુણ]

ટાઈમર/કાઉન્ટર 0 ઓપરેશન મોડની સૂચિ બનાવો અને કોઈપણને સમજાવો..

જલ્દી

Table 5: Timer0 ઓપરેશન મોડ

મોડ	નામ	વર્ણન
0	સામાન્ય	0xFF સુધી ગણતરી, ઓવરફ્લો
1	PWM ફેઝ કરેક્ટ	ફેઝ કરેક્શન સાથે PWM
2	CTC	કંપેર પર ટાઇમર ક્લિયર
3	ફાસ્ટ PWM	ઉચ્ચ ફ્રીક્વન્સી PWM

સામાન્ય મોડ સમજૂતી:

- કાઉન્ટર ઓપરેશન: સતત 0x00 થી 0xFF સુધી ગણતરી કરે છે
- ઓવરફ્લો ફ્લેગ: કાઉન્ટર 0x00 પર ઓવરફ્લો થાય છે ત્યારે TOV0 ફ્લેગ સેટ થાય છે
- ઇન્ટરપ્ટ જનરેશન: ઓવરફ્લો કન્ડિશન પર ઇન્ટરપ્ટ જનરેટ કરી શકે છે

સ્મરણ સહાયક: "સામાન્ય ગણો, PWM પલ્સ કરે, CTC ક્લિયર કરે"

પ્રશ્ન 2(ક) [7 ગુણ]

સ્કેચ સાથે, ATmega32 ની દરેક પિનનું કાર્ય ઓળખો અને લખો.

જલ્દી

આકૃતિ: ATmega32 પિન કોન્ફિગરેશન

[illegible]

(XCK/T0)	PB0	1	40	PA0 (ADC0)
(T1)	PB1	2	39	PA1 (ADC1)
(INT2/AIN0)	PB2	3	38	PA2 (ADC2)
(OC0/AIN1)	PB3	4	37	PA3 (ADC3)
(SS)	PB4	5	36	PA4 (ADC4)
(MOSI)	PB5	6	35	PA5 (ADC5)
(MISO)	PB6	7	34	PA6 (ADC6)
(SCK)	PB7	8	33	PA7 (ADC7)
	RST	9	32	AREF
	VCC	10	31	GND
	GND	11	30	AVCC
	XTAL2	12	29	PC7 (TOSC2)

XTAL1	13	28	PC6 (TOSC1)
(RXD) PD0	14	27	PC5 (TDI)
(TXD) PD1	15	26	PC4 (TD0)
(INT0) PD2	16	25	PC3 (TMS)
(INT1) PD3	17	24	PC2 (TCK)
(OC1B) PD4	18	23	PC1 (SDA)
(OC1A) PD5	19	22	PC0 (SCL)
(ICP1) PD6	20	21	PD7 (OC2)

[illegible]

પિન કાર્યો:

- પોર્ટ A: 8-બિટ ADC ઇનપુટ પિન (PA0-PA7)
- પોર્ટ B: SPI કમ્યુનિકેશન અને ટાઇમર કાર્યો
- પોર્ટ C: JTAG ઇન્ટરફેસ અને I2C કમ્યુનિકેશન
- પોર્ટ D: UART કમ્યુનિકેશન અને બાહ્ય ઇન્ટરપ્ટ
- પાવર પિન: VCC, GND, AVCC એનાલોગ સપ્લાય માટે
- ક્રિસ્ટલ પિન: XTAL1, XTAL2 બાહ્ય ઓસિલેટર માટે

સ્મરણ સહાયક: "એનાલોગ-A, બસ-B, કમ્યુનિકેશન-C, ડેટા-D"

પ્રશ્ન 2(અ અથવા) [3 ગુણ]

ATmega32 ની ડેટા મેમરીની રચના સમજાવો.

જલિયા

Table 6: ATmega32 મેમરી ઓર્ગેનાઈઝેશન

મેમરી પ્રકાર	એડ્રેસ રેન્જ	કદ
રજિસ્ટર	0x00-0x1F	32 બાઇટ
I/O રજિસ્ટર	0x20-0x5F	64 બાઇટ
આંતરિક SRAM	0x60-0x25F	2048 બાઇટ

- સામાન્ય હેતુ રજિસ્ટર: અંગકાણિત ઓપરેશન માટે R0-R31
- I/O મેમરી જગ્યા: પેરિફેરલ માટે કંટ્રોલ રજિસ્ટર
- આંતરિક SRAM: પ્રોગ્રામ એક્ઝિક્યુશન દરમિયાન વેરિએબલ સ્ટોરેજ

સ્મરણ સહાયક: "રજિસ્ટર, I/O, SRAM - RIS"

પ્રશ્ન 2(બ અથવા) [4 ગુણ]

ટાઈમર/કાઉન્ટર 0 ના TIFR અને TCCR રજિસ્ટર દોરો.

જલિયા

આકૃતિ: Timer0 રજિસ્ટર

```

TIFR (
+{-{-}-{-}+{-}-{-}-{-}+{-}-{-}-{-}+{-}-{-}-{-}+{-}-{-}-{-}+{-}-{-}-{-}+{-}-{-}-{-}+
| {- | {-} | {-} | {-} | {-} | OCF2|TOV2|TOV0|OCF0|TOV1|OCF1A|ICF1|OCF1B|}
+{-{-}-{-}+{-}-{-}-{-}+{-}-{-}-{-}+{-}-{-}-{-}+{-}-{-}-{-}+{-}-{-}-{-}+{-}-{-}-{-}+
7 6 5 4 3 2 1 0

```

```

TCCR0 (    /          0)
+{--}{--}{--}+{--}{--}{--}+{--}{--}{--}+{--}{--}{--}+{--}{--}{--}+{--}{--}{--}+{--}{--}{--}+{--}{--}{--}+
|FOC0|WGM00|COM01|COM00|WGM01|  {-  |CS02|CS01|CS00|}
+{--}{--}{--}+{--}{--}{--}+{--}{--}{--}+{--}{--}{--}+{--}{--}{--}+{--}{--}{--}+{--}{--}{--}+{--}{--}{--}+
      7      6      5      4      3      2      1      0

```

બિટ કાર્યો:

- TOV0: Timer0 ઓવરફ્લો ફ્લેગ બિટ
- OCF0: Timer0 આઉટપુટ કંપેર મેચ ફ્લેગ
- CS02:CS00: પ્રીસ્કેલર માટે ક્લોક સિલેક્ટ બિટ
- WGM01:WGM00: વેવફોર્મ જનરેશન મોડ બિટ

સ્મરણ સહાયક: ``TIFR ફ્લેગ બતાવે, TCCR ક્લોક કંટ્રોલ કરે"

પ્રશ્ન 2(ક અથવા) [7 ગુણ]

AVR માઇક્રોકંટ્રોલરનો સામાન્ય બ્લોક ડાયાગ્રામ દોરો અને સમજાવો.

જવાબ

આકૃતિ: AVR આર્કિટેક્ચર



ઘટકો:

- CPU કોર: ઇન્સ્ટ્રક્શન એક્ઝિક્યુટ કરે છે અને સિસ્ટમ ઓપરેશન કંટ્રોલ કરે છે
- પ્રોગ્રામ મેમરી: બિન-અસ્થિર flash માં એપ્લિકેશન કોડ સ્ટોર કરે છે
- ડેટા મેમરી: વેરિએબલ અને સ્ટેક માટે અસ્થાયી સ્ટોરેજ
- ALU: અંકગણિત અને તાર્કિક ઓપરેશન કરે છે
- રજિસ્ટર ફાઇલ: 32 સામાન્ય-હેતુ વર્કિંગ રજિસ્ટર
- I/O સિસ્ટમ: બાહ્ય હાર્ડવેર ઘટકો સાથે ઇન્ટરફેસ
- પેરિફેરલ: બિલ્ટ-ઇન મોડ્યુલ જેમ કે ટાઇમર, UART, ADC

સ્મરણ સહાયક: ``CPU પ્રોગ્રામ, ડેટા, I/O, પેરિફેરલ કંટ્રોલ કરે - CPDIP"

પ્રશ્ન 3(અ) [3 ગુણ]

10 ms વિલંબ સાથે સતત પોર્ટ B ના તમામ બિટ્સને ટોગલ કરવા માટે AVR C પ્રોગ્રામ લખો.

જવાબ

```
\#include {avr/io.h}
\#include {util/delay.h}

int main()
```

```

\{
    DDRB = 0xFF;          //    B

    while(1)
    \{
        PORTB = 0xFF;      //
        \_delay\_ms(10);    // 10ms
        PORTB = 0x00;      //
        \_delay\_ms(10);    // 10ms
    \}
\}

```

મુખ્ય મુદ્દાઓ:

- **DDRB = 0xFF:** પોર્ટ B ના બધા પિનને આઉટપુટ તરીકે કોન્ફિગર કરે છે
- **PORTB ટોગલ:** 0xFF અને 0x00 વચ્ચે બદલાય છે

સ્મરણ સહાયક: ``DDR દિશા, PORT આઉટપુટ``

પ્રશ્ન 3(બ) [4 ગુણ]

MAX232 નું કાર્ય સમજાવો.

જવાબ

Table 7: MAX232 કાર્યો

કાર્ય	વર્ણન
લેવલ કન્વર્ઝન	TTL થી RS232 વોલ્ટેજ લેવલ
ચાર્જ પંપ	+5V સપ્લાયથી $\pm 10V$
લાઇન ડ્રાઇવર	બે ટ્રાન્સમિટ ડ્રાઇવર
લાઇન રિસીવર	બે રિસીવ રિસીવર

- **વોલ્ટેજ કન્વર્ઝન:** 0-5V TTL ને $\pm 12V$ RS232
- **સીરીયલ કમ્યુનિકેશન:** માઇક્રોકન્ટ્રોલરને PC સાથે કમ્યુનિકેટ કરવા સક્ષમ બનાવે છે
- **ડ્યુઅલ ચેનલ:** બે-દિશાવાળી કમ્યુનિકેશનને સમાંતર સપોર્ટ કરે છે

સ્મરણ સહાયક: ``MAX232 માઇક્રોકન્ટ્રોલરને PC સાથે મળાવે છે``

પ્રશ્ન 3(ક) [7 ગુણ]

કેટલાક વિલંબ સાથે સતત PORTC ના તમામ બિટ્સને ટોગલ કરવા માટે AVR C પ્રોગ્રામ લખો. વિલંબ જનરેટ કરવા માટે પ્રીસ્કેલર વિકલ્પ વગર અને ટાઈમર 0, મોડ 0 નો ઉપયોગ કરવો.

જવાબ

```

#include {avr/io.h}

void timer0\_delay()
\{
    TCNT0 = 0;          //
    TCCR0 = 0x01;       //
    while(! (TIFR && (1<}>TOV0))); //
    TIFR |= (1<}>TOV0);   //
    TCCR0 = 0;          //
\}

int main()

```

```

\{
    DDRC = 0xFF;          //    C

    while(1)
    \{
        PORTC = 0xFF;      //
        for(int i=0; i<100; i++)
            timer0\_delay(); //

        PORTC = 0x00;      //
        for(int i=0; i<100; i++)
            timer0\_delay(); //
    \}
\}

```

મુખ્ય લક્ષણો:

- **Timer0 સામાન્ય મોડ:** 0 થી 255 સુધી ગણે છે પછી ઓવરફ્લો
- **કોઈ પ્રીસ્કેલર નહીં:** ટાઇમર સિસ્ટમ ક્લોક સ્પીડે ચાલે છે
- **ઓવરફ્લો ડિટેક્શન:** TOV0 ફ્લેગ ટાઇમર ઓવરફ્લો દર્શાવે છે
- **વિલંબ જનરેશન:** બહુવિધ ટાઇમર ચક્ર દૃશ્યમાન વિલંબ બનાવે છે

સ્મરણ સહાયક: “ટાઇમર ગણે, ઓવરફ્લો ફ્લેગ, વિલંબ જનરેટ કરે”

પ્રશ્ન 3(અ અથવા) [3 ગુણ]

EEPROM ના સ્થાન 0X011F માં #30h સ્ટોર કરવા માટે AVR C પ્રોગ્રામ લખો.

જવાબ

```

#include {avr/io.h}
#include {avr/eeprom.h}

int main()
\{
    eeprom\_write\_byte((uint8\_t*)0x011F, 0x30);
    return 0;
\}

```

વૈકલ્પિક પદ્ધતિ:

```

#include {avr/io.h}

int main()
\{
    while(EECR && (1<EEMWE)); //
    EEAR = 0x011F;           //
    EEDR = 0x30;             //
    EECR |= (1<EEMWE);       //
    EECR |= (1<EEMWE);       //
\}

```

સ્મરણ સહાયક: “એડ્રેસ, ડેટા, માસ્ટર, લેખન - ADMW”

પ્રશ્ન 3(બ અથવા) [4 ગુણ]

C માં AVR પ્રોગ્રામિંગ માટે વિવિધ ડેટા પ્રકારોની ચર્ચા કરો.

Table 8: AVR C ડેટા પ્રકાર

ડેટા પ્રકાર	કદ	રેન્જ
char	1 બાઇટ	-128 થી 127
unsigned char	1 બાઇટ	0 થી 255
int	2 બાઇટ	-32768 થી 32767
unsigned int	2 બાઇટ	0 થી 65535
long	4 બાઇટ	$-2^{31}2^{31} - 1$
float	4 બાઇટ	IEEE 754 ફ્લોટ

- **મેમરી કાર્યક્ષમતા:** સૌથી નાના યોગ્ય ડેટા પ્રકારની પસંદગી કરો
 - **Unsigned પ્રકાર:** જ્યારે નેગેટિવ મૂલ્યોની જરૂર ન હોય ત્યારે ઉપયોગ કરો
 - **Integer અંકગણિત:** ફ્લોટિંગ-પોઇન્ટ ઓપરેશન કરતાં ઝડપી
- સ્મરણ સહાયક:** "મેમરી કાર્યક્ષમતા માટે યોગ્ય કદ પસંદ કરો"

પ્રશ્ન 3(ક અથવા) [7 ગુણ]

સીરીયલ ડેટા ટ્રાન્સમિશન માટે AVR C પ્રોગ્રામ્સ લખો.

```
\#include {avr/io.h}

void uart\_init(unsigned int baud)
\{
    UBRRH = (unsigned char)(baud{}/8);
    UBRRL = (unsigned char)baud;
    UCSRB = (1{}/TXEN); //
    UCSRC = (1{}/URSEL)|(3{}/UCSZ0); // 8{- }
\}

void uart\_transmit(unsigned char data)
\{
    while(!(UCSRA & (1{}/UDRE))); //
    UDR = data; //
\}

void uart\_send\_string(char *str)
\{
    while(*str)
    \{
        uart\_transmit(*str++);
    \}
\}

int main()
\{
    uart\_init(51); // 8MHz 9600 baud

    while(1)
    \{
        uart\_send\_string("Hello World{rn}");
        for(long i=0; i{}/100000; i++); //
    \}
\}
```

મુખ્ય ઘટકો:

- બોડ રેટ સેટિંગ: UBRR રજિસ્ટર કમ્યુનિકેશન સ્પીડ સેટ કરે છે

- ટ્રાન્સમિટ સક્ષમ: TXEN બિટ UART ટ્રાન્સમિટર સક્ષમ કરે છે
 - ડેટા ટ્રાન્સમિશન: UDR રજિસ્ટર ટ્રાન્સમિટ કરવાનો ડેટા હોલ્ડ કરે છે
 - બફર ચેક: UDRE ફ્લેગ ટ્રાન્સમિટ બફર ખાલી દર્શાવે છે
- સ્મરણ સહાયક: "ઇનિટ, સક્ષમ, ચેક, ટ્રાન્સમિટ - IECT"

પ્રશ્ન 4(અ) [3 ગુણ]

ADMUX રજિસ્ટર સમજાવો.

જવાબ

Table 9: ADMUX રજિસ્ટર બિટ્સ

બિટ	નામ	કાર્ય
REFS1:0	રેફરન્સ સિલેક્ટ	વોલ્ટેજ રેફરન્સ પસંદગી
ADLAR	લેફ્ટ એડજસ્ટ	પરિણામ ડાબે એડજસ્ટમેન્ટ
MUX4:0	ચેનલ સિલેક્ટ	ADC ઇનપુટ ચેનલ પસંદગી

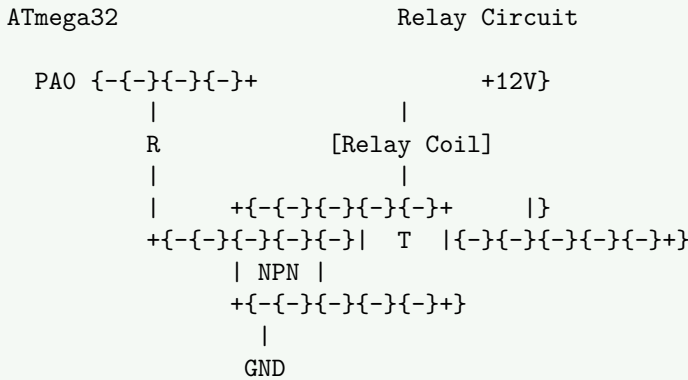
- રેફરન્સ વોલ્ટેજ: આંતરિક/બાહ્ય વોલ્ટેજ રેફરન્સ પસંદ કરે છે
 - પરિણામ ફોર્મેટ: ADLAR બિટ 10-બિટ પરિણામ એલાઇનમેન્ટ એડજસ્ટ કરે છે
 - ચેનલ પસંદગી: MUX બિટ્સ કયા ADC પિનને વાંચવો તે પસંદ કરે છે
- સ્મરણ સહાયક: "રેફરન્સ, એડજસ્ટ, ચેનલ - RAC"

પ્રશ્ન 4(બ) [4 ગુણ]

ATmega32 સાથે ઇન્ટરફેસિંગ રિલે દોરો અને સમજાવો.

જવાબ

આકૃતિ: રિલે ઇન્ટરફેસિંગ



T = BC547 Transistor
R = 1K Resistor

ઘટકો:

- ટ્રાન્ઝિસ્ટર સ્વિચ: BC547 NPN ટ્રાન્ઝિસ્ટર ઇલેક્ટ્રોનિક સ્વિચ તરીકે કામ કરે છે
 - બેઝ રજિસ્ટર: 1KΩ માઇક્રોકંટ્રોલરથી બેઝ કરન્ટ મર્યાદિત કરે છે
 - રિલે કોઇલ: 12V રિલે બાહ્ય હાઇ-પાવર ઉપકરણો ઓપરેટ કરે છે
 - પ્રોટેક્શન ડાયોડ: બેક EMF થી બચાવવા માટે ફ્રીવ્હીલિંગ ડાયોડ
- સ્મરણ સહાયક: "માઇક્રો ટ્રાન્ઝિસ્ટર કંટ્રોલ કરે, ટ્રાન્ઝિસ્ટર રિલે કંટ્રોલ કરે"

પ્રશ્ન 4(ક) [7 ગુણ]

AVR માં TWI રજિસ્ટર દોરો અને સમજાવો.

જવાબ

આકૃતિ: TWI રજિસ્ટર સ્ટ્રક્ચર

TWCR (TWI Control Register)

+{-}{-}{-}{-}{-}{-}+{-}{-}{-}{-}{-}{-}+{-}{-}{-}{-}{-}{-}+{-}{-}{-}{-}{-}{-}+{-}{-}{-}{-}{-}{-}+{-}{-}{-}{-}{-}{-}
|TWINT|TWEA|TWSTA|TWSTO|TWWC|TWEN| {- |TWIE|}
+{-}{-}{-}{-}{-}{-}+{-}{-}{-}{-}{-}{-}+{-}{-}{-}{-}{-}{-}+{-}{-}{-}{-}{-}{-}+{-}{-}{-}{-}{-}{-}+{-}{-}{-}{-}{-}{-}
7 6 5 4 3 2 1 0

TWSR (TWI Status Register)

+{-}{-}{-}{-}{+}{-}{-}{-}{-}{-}{+}{-}{-}{-}{-}{-}{+}{-}{-}{-}{-}{-}{+}{-}{-}{-}{-}{-}{+}{-}{-}{-}{-}{-}{+}{-}{-}{-}{-}{-}{+}{-}{-}{-}{-}{-}
|TWS7|TWS6|TWS5|TWS4|TWS3| {- |TWPS1|TWPS0|}
+{-}{-}{-}{-}{-}{+}{-}{-}{-}{-}{-}{+}{-}{-}{-}{-}{-}{+}{-}{-}{-}{-}{-}{+}{-}{-}{-}{-}{-}{+}{-}{-}{-}{-}{-}{+}{-}{-}{-}{-}{-}{+}{-}{-}{-}{-}{-}
76543210

TWDR (TWI Data Register)

+{-}{-}{-}{-}{-}+{-}{-}{-}{-}{-}+{-}{-}{-}{-}{-}+{-}{-}{-}{-}{-}+{-}{-}{-}{-}{-}+{-}{-}{-}{-}{-}+{-}{-}{-}{-}{-}+{-}{-}{-}{-}{-}
|TWD7|TWD6|TWD5|TWD4|TWD3|TWD2|TWD1|TWD0|
+{-}{-}{-}{-}{-}+{-}{-}{-}{-}{-}+{-}{-}{-}{-}{-}+{-}{-}{-}{-}{-}+{-}{-}{-}{-}{-}+{-}{-}{-}{-}{-}+{-}{-}{-}{-}{-}+{-}{-}{-}{-}{-}
7 6 5 4 3 2 1 0

રજિસ્ટર કાર્યો:

- **TWCR:** TWI ઓપરેશન અને ઇન્ટરપ્ટ હેન્ડલિંગ કંટ્રોલ કરે છે
- **TWSR:** સ્ટેટસ માહિતી અને પ્રીસ્કેલર સેટિંગ પ્રદાન કરે છે
- **TWDR:** ટ્રાન્સમિશન/રિસેપ્શન માટે ડેટા હોલ્ડ કરે છે
- **TWAR:** સ્લેવ તરીકે ઓપરેટ કરતી વખતે સ્લેવ એડ્રેસ સેટ કરે છે
- **TWBR:** TWI કમ્યુનિકેશન માટે બિટ રેટ સેટ કરે છે
- **TWINT:** ઇન્ટરપ્ટ ફ્લેગ 1 લખીને ક્લિયર થાય છે
- **Start/Stop:** TWSTA અને TWSTO I2C કન્ડિશન કંટ્રોલ કરે છે

સ્મરણ સહાયક: "કંટ્રોલ, સ્ટેટસ, ડેટા, એક્સેસ, બિટ રેટ - CSDAB"

પ્રશ્ન 4(અ અથવા) [3 ગુણ]

ADCSRA રજિસ્ટર સમજાવો.

ଉଦାହ

Table 10: ADCSRA રજિસ્ટર બિટ્સ

બિટ	નામ	કાર્ય
ADEN	ADC સક્ષમ	ADC મોડ્યુલ સક્ષમ કરે છે
ADSC	કન્વર્ઝન શરૂ કરો	ADC કન્વર્ઝન શરૂ કરે છે
ADATE	ઓટો ટ્રિગર	ઓટો ટ્રિગર મોડ સક્ષમ કરે છે
ADIF	ઇન્ટરપ્ટ ફ્લેગ	ADC કન્વર્ઝન પૂર્ણ ફ્લેગ
ADIE	ઇન્ટરપ્ટ સક્ષમ	ADC ઇન્ટરપ્ટ સક્ષમ કરે છે
ADPS2:0	પ્રીસ્કેલર	ADC ક્લોક પ્રીસ્કેલર સેટ કરે છે

- **ADC કંટ્રોલ:** ADEN ADC સક્ષમ કરે છે, ADSC કન્વર્ઝન શરૂ કરે છે
- **ઇન્ટરપ્ટ સિસ્ટમ:** કન્વર્ઝન પૂર્ણ થાય ત્યારે ADIF ફ્લેગ સેટ થાય છે

સ્મરણ સહાયક: "સક્ષમ, શરૂ, ટિગર, ઇન્ટરપ્ટ, પ્રીસ્કેલ - ESTIP"

પ્રશ્ન 4(બ અથવા) [4 ગુણ]

ATmega32 સાથે LM35 નું ઇન્ટરફેસિંગ દોરો અને સમજાવો.

જવાબ

આફતિ: LM35 ઇન્ટરફેસિંગ

```
LM35                                     ATmega32
+5V {-{-}{-}{-}{-}+                    }
      |
      [LM35]
      |
GND {-{-}{-}{-}{-}+                    }
      |
Vout {-{-}{-}{-}+{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-} PA0 (ADC0)}

Temperature Sensor
Output: 10mV/~
```

કનેક્શન વિગતો:

- પાવર સપ્લાય: LM35 ને +5V અને ગ્રાઉન્ડ કનેક્શનની જરૂર છે
- આઉટપુટ વોલ્ટેજ: પ્રતિ ડિગ્રી સેલ્સિયસ 10mV ઉત્પન્ન કરે છે
- ADC ઇનપુટ: LM35 આઉટપુટને ADC ચેનલ (PA0) સાથે કનેક્ટ કરો
- ટેમ્પરેચર ગણતરી:
$$= (ADC_Value \times 5000mV) / (1024 \times 10mV)$$

કોડ ઉદાહરણ:

```
float temp = (adc\_read() * 5.0 * 100.0) / 1024.0;
```

સ્મરણ સહાયક: "LM35 પ્રતિ ડિગ્રી 10mV આપે છે"

પ્રશ્ન 4(ક અથવા) [7 ગુણ]

ATmega32 સાથે MAX7221 નો ઉપયોગ કરીને બહુવિધ 7-સેગમેન્ટ ડિસ્પ્લેના ઇન્ટરફેસિંગ દોરો અને સમજાવો.

જવાબ

આફતિ: MAX7221 ઇન્ટરફેસિંગ

```
ATmega32          MAX7221          7{-Segment Displays}

PB5(MOSI) {-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-} DIN           DIG0 {-}{-}{-}{-}{-} Display 1}
PB7(SCK)  {-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-} CLK         DIG1 {-}{-}{-}{-}{-} Display 2 }
PB4(SS)   {-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-} CS          DIG2 {-}{-}{-}{-}{-} Display 3}

                                DIG3 {-}{-}{-}{-}{-} Display 4}
+5V {-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-} VCC             DIG4 {-}{-}{-}{-}{-} Display 5}
GND {-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-} GND            DIG5 {-}{-}{-}{-}{-} Display 6}

                                DIG6 {-}{-}{-}{-}{-} Display 7}

SEGA {-}{-}{-}{-}{-} Common segments}
SEGB      to all displays
SEGC
SEGD
SEGE
SEGF
SEGG
SEGDP
```

લક્ષણો:

- **SPI કમ્યુનિકેશન:** કંટ્રોલ માટે સીરીયલ પેરિફરલ ઇન્ટરફેસ ઉપયોગ કરે છે
- **બહુવિધ ડિસ્પ્લે:** 8 સુધી સેવન-સેગમેન્ટ ડિસ્પ્લે કંટ્રોલ કરે છે
- **ઓટોમેટિક સ્કેનિંગ:** MAX7221 મલ્ટિપ્લેક્સિંગ ઓટોમેટિક હેન્ડલ કરે છે

- ### મુખ્ય રજિસ્ટર:

- સ્મરણ સહાયક:** “SPI બહુવિધ ડિસ્ખલે માટે સીરીયલ ડેટા મોકલે છે”

SPCR રજિસ્ટર સમજાવો.

Table 11: SPCR રજિસ્ટર બિટ્સ

સ્મરણ સહાયક: "ઇન્ટરપ્ટ, સક્ષમ, ડેટા, માસ્ટર, ક્લોક સેટિંગ્સ - IEDMC"

ATmega32	L293D	DC Motor
PA0 {-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}	IN1	OUT1 {-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{+}
PA1 {-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}	IN2	OUT2 {-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{+} [Motor]}
		M
+5V {-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}	VCC1	VCC2 {-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{+12V }
GND {-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}	GND	GND {-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{+12V }
PA2 {-}{-}{-}{-}{-}{-}{-}{-}{-}{-}{-}	EN1	}
	Input Logic Table:	
	IN1 IN2 Motor	
	0 0 Stop	
	0 1 CCW	
	1 0 CW	
	1 1 Brake	

12

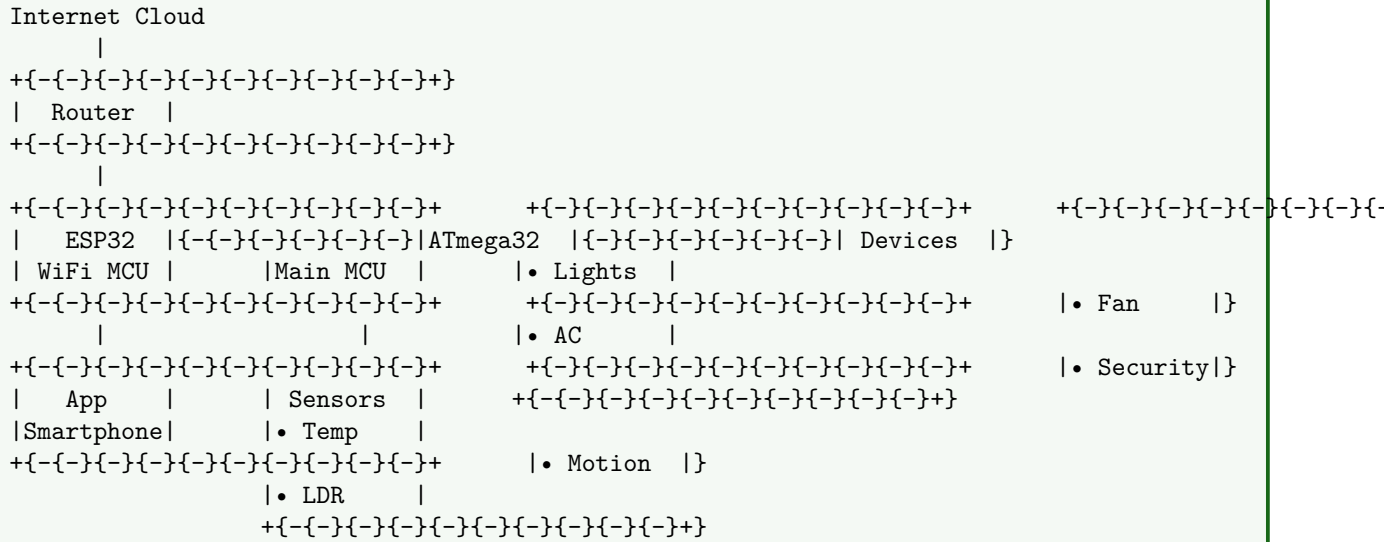
- L293D ડ્રાઇવર: મોટર કંટ્રોલ માટે કરન્ટ એમ્પ્લિફિકેશન પ્રદાન કરે છે
 - પાવર સપ્લાય: લોજિક માટે +5V, મોટર પાવર માટે +12V
 - કંટ્રોલ સિગ્નલ: IN1, IN2 મોટરની દિશા નક્કી કરે છે
 - સક્ષમ પિન: EN1 મોટર ઓન/ઓફ અને સ્પીડ (PWM) કંટ્રોલ કરે છે
- સ્મરણ સહાયક: “લોજિક દિશા કંટ્રોલ કરે, સક્ષમ સ્પીડ કંટ્રોલ કરે”

પ્રશ્ન 5(ક) [7 ગુણ]

IoT આધારિત હોમ ઓટોમેશન સિસ્ટમ સમજાવો.

જવાબ

આકૃતિ: IoT હોમ ઓટોમેશન સિસ્ટમ



સિસ્ટમ ઘટકો:

- ઇન્ટરનેટ કનેક્ટિવિટી: WiFi મોડ્યુલ સિસ્ટમને ઇન્ટરનેટ સાથે કનેક્ટ કરે છે
- મોબાઇલ એપ્લિકેશન: રિમોટ કંટ્રોલ અને મોનિટરિંગ માટે યુઝર ઇન્ટરફેસ
- સેન્સર નેટવર્ક: ઓટોમેશન માટે ટેમ્પરેચર, મોશન, લાઇટ સેન્સર
- કંટ્રોલ ઉપકરણો: રિલે ઘરના ઉપકરણો અને લાઇટ કંટ્રોલ કરે છે
- સેન્ટ્રલ કંટ્રોલર: માઇક્રોકન્ટ્રોલર કમાન્ડ અને સેન્સર ડેટા પ્રોસેસ કરે છે
- ક્લાઉડ સેવાઓ: ડેટા સ્ટોર કરે છે અને રિમોટ એક્સેસ સક્ષમ કરે છે

લક્ષણો:

- રિમોટ કંટ્રોલ: ઇન્ટરનેટ દ્વારા ગમે ત્યાંથી ઉપકરણો કંટ્રોલ કરો
- ઓટોમેશન: સેન્સર રીડિંગ આધારે ઓટોમેટિક કંટ્રોલ
- એનર્જી સેવિંગ: સ્માર્ટ શેડ્યુલિંગ પાવર વપરાશ ઘટાડે છે
- સુરક્ષા મોનિટરિંગ: સુરક્ષા માટે મોશન સેન્સર અને કેમેરા
- ડેટા લૉગિંગ: વિશ્લેષણ માટે ઐતિહાસિક ડેટા સ્ટોરેજ

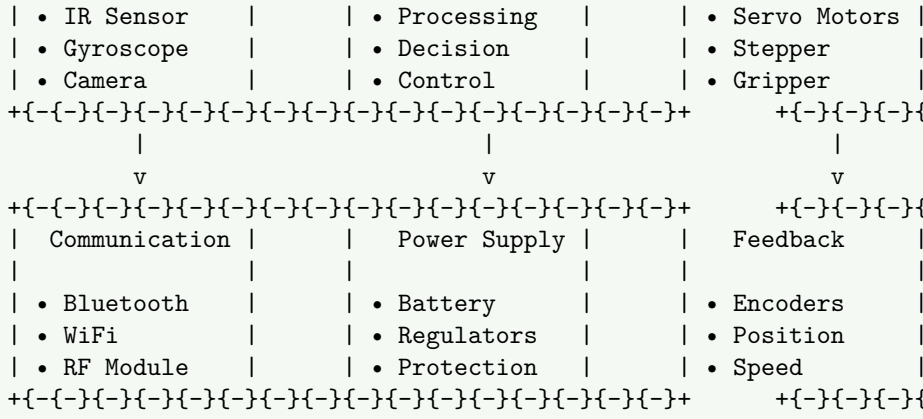
સ્મરણ સહાયક: “ઇન્ટરનેટ ફોનને ઘરના ઉપકરણો સાથે જોડે છે - IPHD”

પ્રશ્ન 5(અ અથવા) [3 ગુણ]

SPSR રજિસ્ટર સમજાવો.

જવાબ

Table 12: SPSR રજિસ્ટર બિટ્સ



સિસ્ટમ ઘટકો:

Table 13: રોબોટિક્સ સિસ્ટમ એલિમેન્ટ્સ

ઘટક	કાર્ય	ઉદાહરણો
સેન્સર	પર્યાવરણ સેન્સિંગ	અલ્ટ્રાસોનિક, IR, કેમેરા
કંટ્રોલર	નિર્ણય લેવો	ATmega32, Arduino
એક્ઝ્યુટેટર	ભૌતિક હલનચલન	મોટર, સર્વો
કમ્યુનિકેશન	રિમોટ કંટ્રોલ	બ્લૂટૂથ, WiFi
પાવર	એનર્જી સપ્લાય	બેટરી, રેગ્યુલેટર
ફીડબેક	પોઝિશન સેન્સિંગ	એન્કોડર, જાયરોસ્કોપ

કંટ્રોલ અલ્ગોરિધમ:

- **સેન્સર:** સેન્સર ઉપયોગ કરીને પર્યાવરણથી ડેટા એકત્રિત કરો
- **પ્રોસેસ:** સેન્સર ડેટાનું વિશ્લેષણ કરો અને નિર્ણયો લો
- **એક્ટ:** નિર્ણયો આધારે મોટર અને એક્ઝ્યુટેટર કંટ્રોલ કરો
- **ફીડબેક:** વાસ્તવિક હલનચલન મોનિટર કરો અને કંટ્રોલ એડજસ્ટ કરો
- **કમ્યુનિકેટ:** સ્ટેટસ મોકલો અને વાયરલેસ કમાન્ડ રિસીવ કરો

એપ્લિકેશન:

- **સ્વાયત્ત નેવિગેશન:** રોબોટ સેન્સર ઉપયોગ કરીને સ્વતંત્ર રીતે મૂવ કરે છે
- **ઓબ્જેક્ટ મેનિપ્યુલેશન:** પિક અને પ્લેસ કાર્યો માટે ગ્રિપર કંટ્રોલ
- **રિમોટ ઓપરેશન:** વાયરલેસ કમ્યુનિકેશન દ્વારા મેન્યુઅલ કંટ્રોલ
- **પાથ ફોલોવિંગ:** લાઇન ફોલોવિંગ અથવા પૂર્વનિર્ધારિત રૂટ નેવિગેશન
- **ઓબ્સ્ટેકલ એવોઇડન્સ:** અવરોધોની આસપાસ ડાયનેમિક પાથ પ્લાનિંગ

પ્રોગ્રામિંગ સ્ટ્રક્ચર:

```

while(1) \{
    read\_sensors();
    process\_data();
    make\_decision();
    control\_motors();
    check\_feedback();
    communicate\_status();
\}
  
```

સ્મરણ સહાયક: "સેન્સ, પ્રોસેસ, એક્ટ, ફીડબેક, કમ્યુનિકેટ - SPACF"