

Embedded System & Microcontroller Application (4351102) - Summer 2025 Solution Gujarati

Milav Dabgar

May 12, 2025

પ્રશ્ન 1(અ) [3 ગુણ]

ATmega32 ની લાક્ષણિકતાઓ જણાવો.

જવાબ

ATmega32 લાક્ષણિકતાઓ:

કોષ્ટક 1. ATmega32 લાક્ષણિકતાઓ

લાક્ષણિકતા	વર્ણન
Flash Memory	32KB programmable memory
SRAM	2KB internal SRAM
EEPROM	1KB non-volatile data storage
I/O Pins	32 programmable I/O lines
Timers	3 flexible timer/counters
ADC	10-bit 8-channel ADC

વધારાની વિશેષતાઓ:

- **Operating Voltage:** 2.7V થી 5.5V રેન્જ
- **Clock Speed:** 16 MHz સુધી ની ઓપરેશન
- **Communication:** USART, SPI, I2C interfaces

મેમરી ટ્રીક

"Fast SRAM Enjoys Input Timers And Communication"

પ્રશ્ન 1(બ) [4 ગુણ]

માઈક્રોકંટ્રોલર પસંદ કરવા માટેના માપદંડો લખો.

જવાબ

પસંદગીના માપદંડો:

કોષ્ટક 2. પસંદગીના માપદંડો

માપદંડ	વિચારણા
Processing Speed	Clock frequency આવશ્યકતાઓ
Memory Size	Program અને data storage જરૂરિયાત
I/O Requirements	જરૂરી pins ની સંખ્યા
Power Consumption	Battery life વિચારણાઓ
Cost	Budget મર્યાદાઓ
Development Tools	Compiler અને debugger ઉપલબ્ધતા

સિસ્ટમ વિચારણાઓ:

- **Application Type:** Real-time vs general purpose
- **Communication Needs:** Serial, parallel, wireless protocols
- **Package Size:** Final product માં space constraints

મેમરી ટ્રીક

“Processing Memory I/O Power Cost Development Application Communication Package”

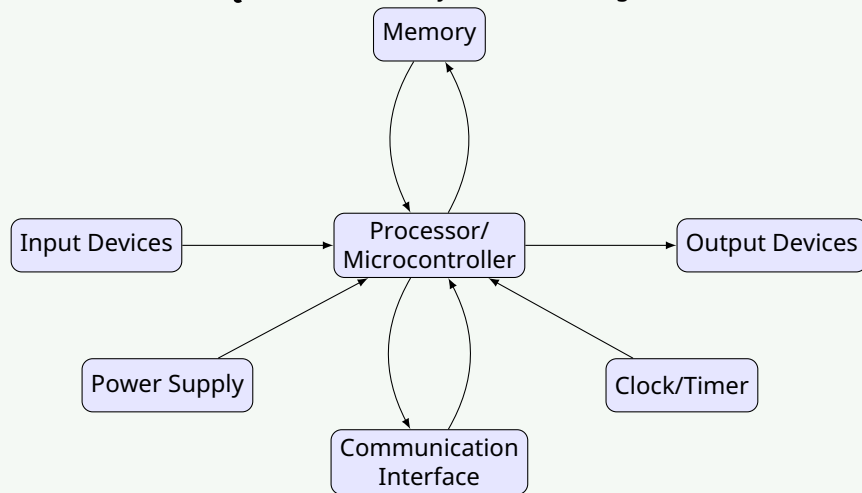
પ્રશ્ન 1(ક) [7 ગુણ]

એમ્બેડેડ સિસ્ટમનો સામાન્ય બ્લોક ડાયાગ્રામ દોરો અને સમજાવો.

જવાબ

સામાન્ય બ્લોક ડાયાગ્રામ:

આકૃતિ 1. Embedded System Block Diagram



બ્લોક કાર્યો:

- **Processor:** Instructions execute કરતું central processing unit
- **Memory:** Program code અને data temporary store કરે છે
- **Input Devices:** Sensors, switches જે system input આપે છે
- **Output Devices:** Actuators, displays જે results બતાવે છે
- **Communication:** External device connectivity માટે interfaces
- **Power Supply:** બધા components ને stable voltage પૂરું પાડે છે
- **Clock/Timer:** System operations અને timing synchronize કરે છે

મેમરી ટ્રીક

"Processors Memory Input Output Communication Power Clock"

OR

પ્રશ્ન 1(ક) [7 ગુણ]

રીયલ ટાઈમ ઓપરેટિંગ સિસ્ટમને વ્યાખ્યાયિત કરો અને તેની લાક્ષણિકતાઓ સમજાવો.

જવાબ

Real Time Operating System (RTOS): કસ્ટ time constraints માં data અને events process કરવા માટે design કરેલું operating system.

કોષ્ટક 3. RTOS લાક્ષણિકતાઓ

લાક્ષણિકતા	વર્ણન
Deterministic	Predictable response times
Preemptive	Higher priority tasks interrupt lower ones
Multitasking	Multiple tasks concurrently run થાય છે
Fast Context Switch	Quick task switching capability
Priority Scheduling	Tasks priority પર આધારે execute થાય છે
Interrupt Handling	Efficient interrupt processing

મુખ્ય વિભાવનાઓ:

- **Hard Real-time:** Deadline miss થવાથી system failure થાય છે
- **Soft Real-time:** Deadline miss થવાથી performance degraded થાય છે
- **Time Constraints:** Operations deadline માં complete થવા જોઈએ

મેમરી ટ્રીક

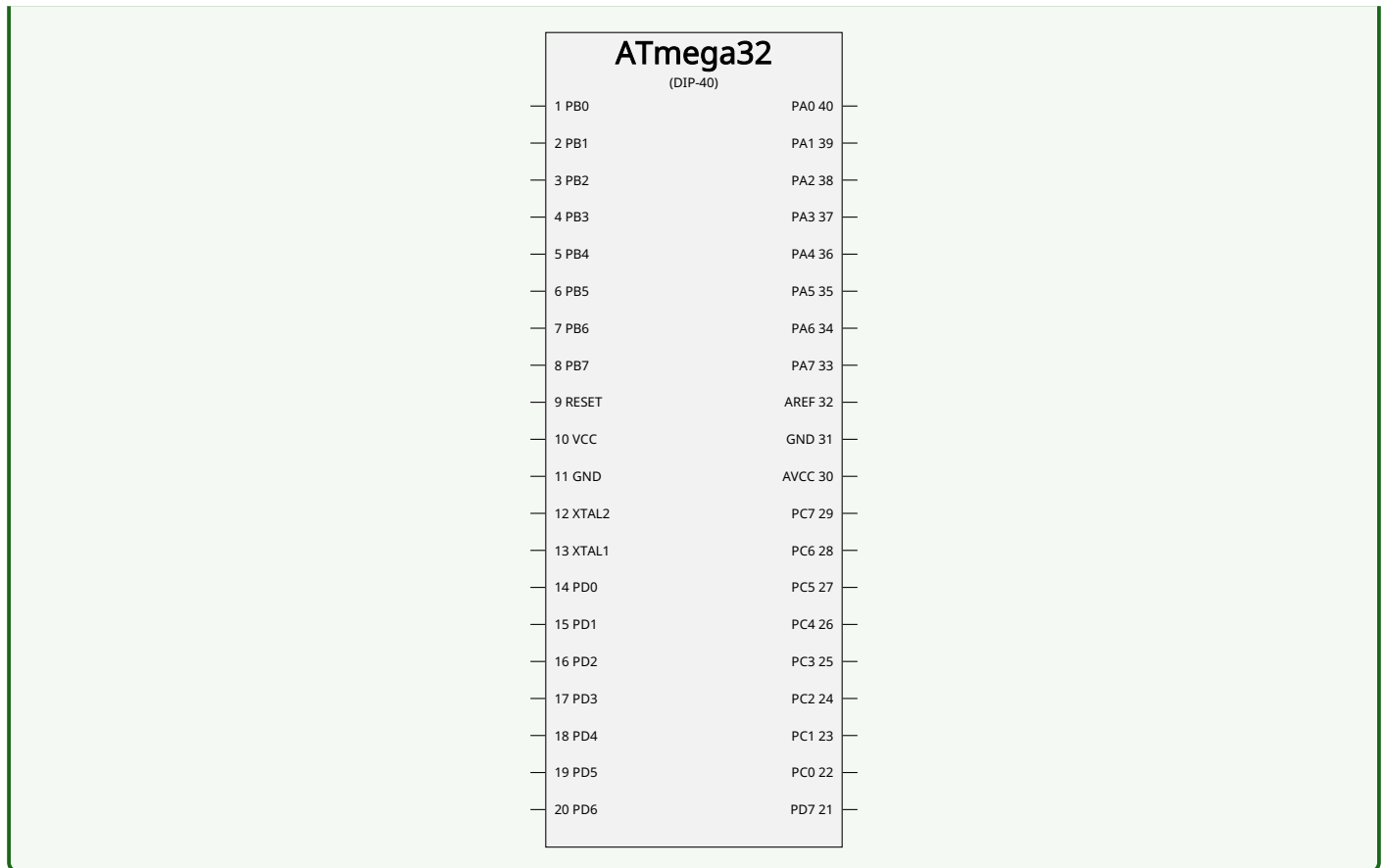
"Deterministic Preemptive Multitasking Fast Priority Interrupt"

પ્રશ્ન 2(અ) [3 ગુણ]

ATmega32 નો પીન ડાયાગ્રામ દોરો.

જવાબ

ATmega32 Pin Configuration:



મેમરી ટ્રીક

``Port B A Reset Vcc Ground Crystal Port D C''

પ્રશ્ન 2(બ) [4 ગુણ]

ATmega32 નો સ્ટેટસ રજિસ્ટર સમજાવો.

જવાબ

Status Register (SREG):

કોષ્ટક 4. SREG Bit Configuration

Bit	Name	Function
Bit 7	I	Global Interrupt Enable
Bit 6	T	Bit Copy Storage
Bit 5	H	Half Carry Flag
Bit 4	S	Sign Bit
Bit 3	V	Overflow Flag
Bit 2	N	Negative Flag
Bit 1	Z	Zero Flag
Bit 0	C	Carry Flag

કાર્યો:

- **Flags Update:** ALU operations દ્વારા automatically set/clear થાય છે

- **Conditional Branching:** Program flow control માટે વપરાય છે
- **Interrupt Control:** I-bit બધા interrupts ને enable/disable કરે છે

મેમરી ટ્રીક

"I Think Half Sign Overflow Negative Zero Carry"

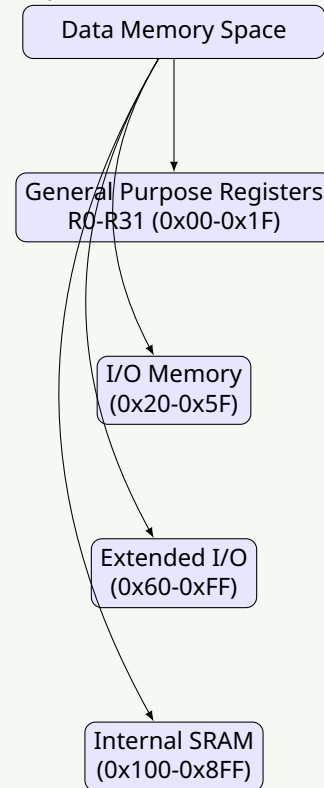
પ્રશ્ન 2(ક) [7 ગુણ]

ATmega32 ની ડેટા મેમરી વિગતવાર સમજાવો.

જવાબ

Data Memory Organization:

આકૃતિ 2. Data Memory Space



મેમરી વિભાગો:

- **General Purpose Registers:** Data operations માટે 32 registers (R0-R31)
- **I/O Memory:** Peripheral control registers (Port, Timer, UART) માટે 64 registers
- **Extended I/O:** Additional peripheral registers અને stack pointer
- **Internal SRAM:** Variables અને stack માટે 2KB volatile memory
- **Stack:** High memory addresses થી low addresses તરફ downward grows થાય છે

મેમરી ટ્રીક

"General I/O Extended SRAM Address Stack"

OR

પ્રશ્ન 2(અ) [3 ગુણ]

DDRx, PINx અને PORTx રજિસ્ટરના કાર્યો લખો.

જવાબ

I/O Port Registers:

કોષ્ટક 5. I/O Registers

Register	Function
DDRx	Data Direction Register - pin direction configure કરે છે
PINx	Pin Input Register - current pin logical state read કરે છે
PORTx	Port Output Register - data/pull-up control write કરે છે

ઓપરેશન્સ:

- **DDRx:** Output માટે 1, Input માટે 0 set કરો
- **PINx:** Physical pin voltage level access કરે છે
- **PORTx:** DDRx=1 હોય ત્યારે High/Low drive કરે છે. DDRx=0 હોય ત્યારે internal Pull-up enable કરે છે

મેમરી ટ્રીક

``Direction Input Output``

OR

પ્રશ્ન 2(બ) [4 ગુણ]

AVR માં EEPROM સાથે સંકળાયેલા વિવિધ I/O રજિસ્ટરો સમજાવો.

જવાબ

EEPROM Registers:

કોષ્ટક 6. EEPROM Registers

Register	Function
EEAR	EEPROM Address Register (High/Low bytes)
EEDR	EEPROM Data Register
EECR	EEPROM Control Register

EECR Control Bits:

- **EERIE:** EEPROM Ready Interrupt Enable
- **EEMWE:** EEPROM Master Write Enable
- **EEWE:** EEPROM Write Enable (Strobe to write)
- **EEERE:** EEPROM Read Enable

પ્રક્રિયા: Write માટે, કોઈ write progress માં નથી તેની ખાતરી કરો, address અને data set કરો, EEMWE set કરો, પછી 4 clock cycles માં EEWE set કરો.

મેમરી ટ્રીક

``Address Data Control Ready Master Write Read``

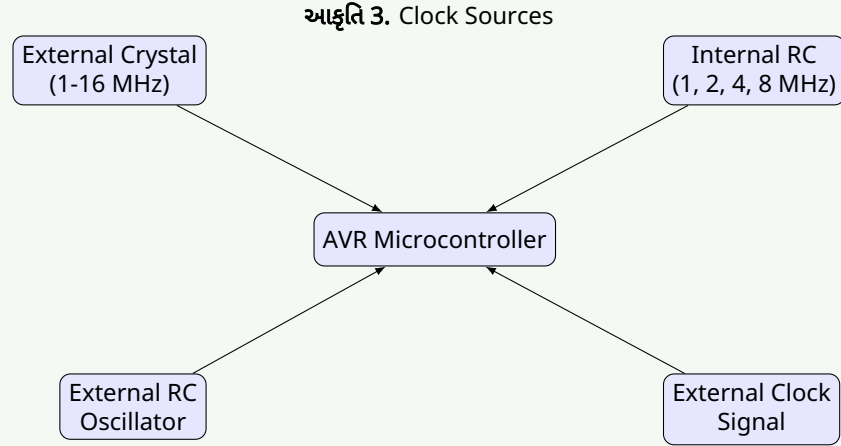
OR

પ્રશ્ન 2(ક) [7 ગુણ]

ક્લોક સોર્સને AVR સાથે જોડવાની વિવિધ રીતો સમજાવો.

જવાબ

Clock Source Options:



કોષ્ટક 7. Clock Source Types

Source	વર્ણન
External Crystal	XTAL1/XTAL2 સાથે જોડાયેલ High precision quartz crystal. Capacitor જરૂરી.
External RC	Resistor-Capacitor circuit. ઓછી કિંમત, ઓછી ચોકસાઈ.
Internal RC	Built-in oscillator. કોઈ external components નહીં. 1MHz default, 8MHz સુધી.
External Clock	External logic signal થી XTAL1 drive કરો.

Configuration:

- **Fuse Bits:** CKSEL3:0 અને SUT1:0 active clock source અને startup time નક્કી કરે છે
- **ઉપયોગ:** Timing-critical (UART) માટે Crystal, cost/power saving માટે Internal RC

મેમરી ટ્રીક

“Crystal RC Internal External Fuse Startup Frequency Components”

પ્રશ્ન 3(અ) [3 ગુણ]

Timer 1 સાથે સંકળાયેલા રજિસ્ટરોનું કાર્ય લખો.

જવાબ

Timer 1 Registers (16-bit):

કોષ્ટક 8. Timer 1 Registers

Register	Function
TCNT1	Timer/Counter Register - Holds current count value (16-bit)
TCCR1A/B	Control Registers - Sets mode, prescaler, compare output
ICR1	Input Capture Register - Captures TCNT1 value on event
OCR1A/B	Output Compare Registers - Values to compare with TCNT1
TIMSK	Interrupt Mask - Enables Overflow/Compare/Capture interrupts
TIFR	Interrupt Flag - Indicates pending interrupts

મેમરી ટ્રીક

“Timer Control Input Output Mask Flag”

પ્રશ્ન 3(બ) [4 ગુણ]

Timer 0 ને સામાન્ય મોડમાં પ્રોગ્રામ કરવાના મુદ્દાઓ લખો.

જવાબ

Normal Mode માટે Programming Steps:

1. **Configure Mode:** Normal Mode select કરવા TCCR0 માં write કરો (WGM01:0 = 00)
2. **Load Initial Value:** TCNT0 માં starting count write કરો
3. **Interrupts:** જો interrupts વાપરતા હોય, TIMSK માં TOIE0 set કરો અને global interrupts enable કરો (sei)
4. **Start Timer:** TCCR0 માં prescaler bits (CS02:0) set કરો
5. **Monitor/Handle:** TIFR માં TOV0 flag માટે wait કરો (polling) અથવા ISR માં handle કરો

```

1 TCCR0 = 0x00;    // Stop timer
2 TCNT0 = 0x00;    // Clear count
3 TCCR0 = 0x01;    // Start: Normal mode, No prescaler
4 while(!(TIFR & 1)); // Wait for overflow
5 TIFR = 0x01;    // Clear flag

```

મેમરી ટ્રીક

“Set Select Load Enable Start”

પ્રશ્ન 3(ક) [7 ગુણ]

ડેટા બાઈટોને સીરીયલ રીસીવ કરાવવા અને તેમને PORTA પર મૂકવા માટેનો C પ્રોગ્રામ લખો. બાઉડ રેટ 9600, 8-બીટ, અને 1-બીટ સેટ કરો.

જવાબ

પ્રોગ્રામ:

```

1 #include <avr/io.h>
2
3 void USART_Init() {
4     // Set baud rate to 9600 (assuming 8MHz clock)
5     UBRRH = 0x00;
6     UBRL = 51;
7 }

```



```

8 // Enable receiver
9 UCSRB = (1<<RXEN);

10
11 // Set frame format: 8 data, 1 stop bit
12 // URSEL must be 1 to write to UCSRC
13 UCSRC = (1<<URSEL) | (3<<UCSZ0);
14 }

15
16 unsigned char USART_Receive() {
17 // Wait for data to be received (RXC flag)
18 while(!(UCSRA & (1<<RXC)));
19 // Get and return received data from buffer
20 return UDR;
21 }

22
23 int main() {
24 DDRA = 0xFF; // Configure PORTA as output
25 USART_Init(); // Initialize USART
26
27 while(1) {
28 PORTA = USART_Receive(); // Receive and display on Port A
29 }
30 return 0;
31 }

```

મેમરી ટ્રીક

``Initialize Receive Display Loop"

OR

પ્રશ્ન 3(અ) [3 ગુણ]

AVR માં સીરીયલ કોમ્યુનિકેશન સાથે સંકળાયેલા રજિસ્ટરોના કાર્યો લખો.

જવાબ

USART Registers:

કોષ્ટક 9. USART Registers

Register	Function
UDR	USART Data Register - Buffer for Tx/Rx data
UCSRA	Control/Status A - Flags (RXC, TXC, UDRE), error bits
UCSRB	Control/Status B - Enable Rx/Tx, Interrupts (RXCIE), 9th bit
UCSRC	Control/Status C - Frame format (Data bits, Parity, Stop bits)
UBRR	Baud Rate Register - Sets communication speed

મેમરી ટ્રીક

``Data Control Status Baud"

OR

પ્રશ્ન 3(બ) [4 ગુણ]

ડેટા સીરીયલ ટ્રાન્સફર કરવા માટે AVR ને પ્રોગ્રામ કરવાના મુદ્દાઓની ચર્ચા કરો.

જવાબ

Serial Transmission Steps:

1. **Baud Rate:** Calculate અને UBRRH/Low registers માં load કરો
2. **Frame Format:** UCSRC માં data bits, parity, અને stop bits set કરો (e.g., 8N1)
3. **Enable Tx:** Transmitter enable કરવા UCSRB માં TXEN bit set કરો
4. **Wait for Buffer:** UDR empty છે તેની ખાતરી કરવા UCSRA માં UDRE flag monitor કરો
5. **Send Data:** Transmission શરૂ કરવા UDR register માં byte write કરો

```
1 void USART_Transmit(unsigned char data) {
2     // Wait for empty transmit buffer
3     while(!(UCSRA & (1<<UDRE)));
4     // Put data into buffer, sends the data
5     UDR = data;
6 }
```

મેમરી ટ્રીક

“Baud Enable Format Wait Load”

OR

પ્રશ્ન 3(ક) [7 ગુણ]

દર 2 મિલિસેકન્ડે માત્ર PORTB.4 બીટને સતત ટોગલ કરવા માટેનો C પ્રોગ્રામ લખો. Delay જનરેટ કરવા timer 1 ને પ્રીસ્કેલર વગર નોર્મલ મોડમાં ઉપયોગ કરો. XTAL=8MHz ધારો.

જવાબ

Timer 1 Delay ગણતરી:

- Target Delay: 2 ms
- Frequency: 8 MHz
- Machine Cycle: $1/8\text{MHz} = 0.125\mu\text{s}$
- Counts Required: $2\text{ms}/0.125\mu\text{s} = 16000 \text{ counts}$
- Timer Start Value: $65536 - 16000 = 49536 (0xC180)$

પ્રોગ્રામ:

```
1 #include <avr/io.h>
2 #include <avr/interrupt.h>
3
4 // ISR for Timer1 Overflow
5 ISR(TIMER1_OVF_vect) {
6     // Toggle PORTB.4
7     PORTB ^= (1<<4);
8
9     // Reload Timer for 2ms
10    TCNT1 = 49536;
11 }
12
13 int main() {
14     // Set PORTB.4 as output
15     DDRB |= (1<<4);
```

```

16
17 // Initialize Timer1
18 // Normal Mode (WGM13:0 = 0000)
19 TCCR1A = 0x00;
20
21 // Start Timer with No Prescaler (CS12:0 = 001)
22 TCCR1B = (1<<CS10);
23
24 // Load initial count
25 TCNT1 = 49536;
26
27 // Enable Timer1 Overflow Interrupt
28 TIMSK |= (1<<TOIE1);
29
30 // Enable Global Interrupts
31 sei();
32
33 while(1) {
34     // Main loop does nothing, ISR handles toggling
35 }
36 return 0;
37 }

```

મેમરી ટ્રીક

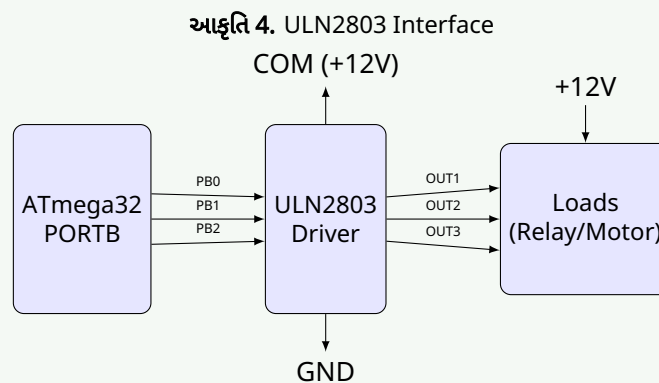
“Configure Timer Calculate Enable Loop”

પ્રશ્ન 4(a) [3 ગુણ]

ULN2803 નો ATmega32 સાથે ઇન્ટરફેસિંગ ડાયાગ્રામ દોરો.

જવાબ

ULN2803 ઇન્ટરફેસિંગ:



કનેક્શન્સ:

- **Inputs:** ATmega32 pins ને સીધા ULN2803 inputs (1B-8B) સાથે connect કરો.
- **Outputs:** Open collector outputs positive supply સાથે જોડાયેલા loads ને drive કરે છે.
- **Common:** Pin 10 ને COM (Load Supply) સાથે (flyback diodes માટે), Pin 9 ને GND સાથે.

મેમરી ટ્રીક

“Input Output Common Supply Ground”

પ્રશ્ન 4(b) [4 ગુણ]

Port B પરથી ડેટા બાઈટ લેવો અને તેને Port C પર મોકલવા AVR માટેનો C પ્રોગ્રામ લખો.

જવાબ

પ્રોગ્રામ:

```

1 #include <avr/io.h>
2
3 int main() {
4     // Configure Directions
5     DDRB = 0x00; // Port B Input તરીકે
6     DDRC = 0xFF; // Port C Output તરીકે
7
8     // Input Port પર Pull-ups Enable કરો
9     PORTB = 0xFF;
10
11     while(1) {
12         // PINB register થી data read કરો
13         unsigned char data = PINB;
14
15         // PORTC register પર data write કરો
16         PORTC = data;
17     }
18     return 0;
19 }

```

- **Direction:** DDRB=0 (In), DDRC=1 (Out).
- **Pull-ups:** DDRB=0 હોય ત્યારે PORTB માં 1 write કરવાથી internal pull-ups enable થાય છે.
- **Transfer:** Continuous loop PINB reads કરે છે અને PORTC updates કરે છે.

મેમરી ટ્રીક

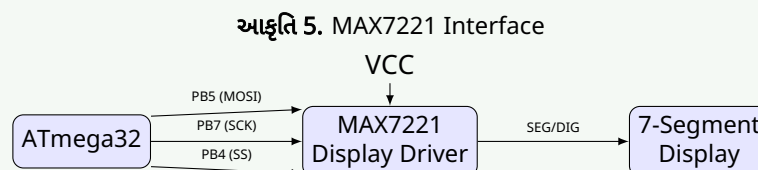
“Configure Enable Read Write”

પ્રશ્ન 4(ક) [7 ગુણ]

MAX7221 નો ATmega32 સાથે ઇન્ટરફેસિંગ ડાયાગ્રામ દોરો અને સમજાવો.

જવાબ

MAX7221 Interface:



Interface વિગતો:

- **Protocol:** SPI (Serial Peripheral Interface). 3-wire: Data (DIN), Clock (CLK), Load (CS).
- **Function:** 8 common-cathode 7-segment displays સુધી drive કરે છે.
- **Multiplexing:** Scanning circuitry internally handle કરે છે.
- **Current Control:** External resistor (ISET pin) દ્વારા set થાય છે.

Programming:

1. LOAD (CS) Low કરો.
2. CLK rising edge પર DIN દ્વારા 16-bit word (Address + Data) shift in કરો.
3. Data latch કરવા LOAD High pulse આપો.

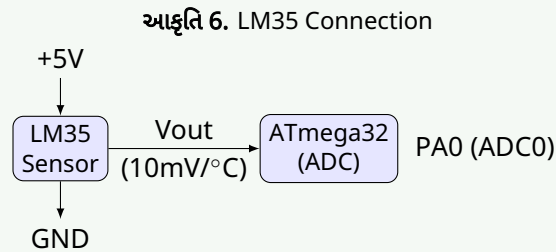
મેમરી ટ્રીક

“SPI Data Clock Load Multiplex Current Program”

OR

પ્રશ્ન 4(અ) [3 ગુણ]

LM35 નો ATmega32 સાથે ઇન્ટરફેસિંગ ડાયાગ્રામ દોરો.

જવાબ**LM35 Interface Circuit:****Specifications:**

- **Output:** Temperature ના પ્રમાણમાં Linear voltage (10mV/°C).
- **Connection:** Vout કોઈપણ ADC channel (e.g., PA0) સાથે connect કરો.
- **Conversion:** $V_{in} \times \frac{1024}{V_{ref}} = \text{ADC Value.}$

મેમરી ટ્રીક

“VCC OUT GND Linear”

OR

પ્રશ્ન 4(બ) [4 ગુણ]

Port C ના બીટ-5 ને મોનીટર કરો, જો તે HIGH હોય તો 55H ને Port B પર મોકલો નહીં તો AAH ને Port B પર મોકલવા AVR માટેનો C પ્રોગ્રામ લખો.

જવાબ**પ્રોગ્રામ:**

```
1 #include <avr/io.h>
```

```

2
3 int main() {
4     DDRC &= ~(1<<5); // PC5 Input તરીકે
5     PORTC |= (1<<5); // PC5 પર Pull-up Enable કરો
6     DDRB = 0xFF; // Port B Output તરીકે
7
8     while(1) {
9         // PC5 High છે કે નહીં check કરો
10        if(PINC & (1<<5)) {
11            PORTB = 0x55; // Pattern 01010101
12        } else {
13            PORTB = 0xAA; // Pattern 10101010
14        }
15    }
16    return 0;
17 }

```

Logic:

- **Masking:** PINC & (1<<5) bit 5 ને isolate કરે છે.
- **Condition:** જો result non-zero (True) હોય, તો bit High છે.

મેમરી ટ્રીક

“Monitor Conditional Output Loop”

OR

પ્રશ્ન 4(ક) [7 ગુણ]

AVR માં SPI ને પ્રોગ્રામ કરવા માટેના રજિસ્ટરોની ચર્ચા કરો.

જવાબ**SPI Registers:**

કોષ્ટક 10. SPI Registers

Register	Function
SPCR	Control Register - EN, Int, Order, Master/Slave, Mode
SPSR	Status Register - Interrupt Flag (SPIF), Collision (WCOL)
SPDR	Data Register - Transmission/Reception માટે Shift register

SPCR Bits:

- **SPE:** SPI Enable.
- **MSTR:** Master (1) અથવા Slave (0) Select.
- **SPR1:0:** Clock Rate Select ($f_{osc}/4$, $/16$, $/64$, $/128$).
- **CPOL/CPHA:** Clock Polarity અને Phase (Mode 0-3).

ઉપયોગ: SPDR માં write કરવાથી transmission શરૂ થાય છે (Master). SPIF set થયા પછી SPDR read કરવાથી received data મળે છે.

મેમરી ટ્રીક

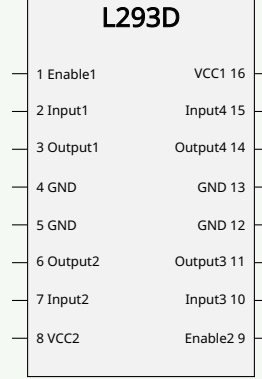
“Control Status Data Configure Enable Write Wait Read”

પ્રશ્ન 5(અ) [3 ગુણ]

મોટર ડ્રાઇવર આઈસી L293D નો પીન ડાયાગ્રામ દોરો.

જવાબ

L293D Pinout:



મેમરી ટ્રીક

``Enable Input Output Supply Ground``

પ્રશ્ન 5(બ) [4 ગુણ]

ADMUX રજિસ્ટર દોરો અને સમજાવો.

જવાબ

ADMUX (ADC Multiplexer Selection Register):

કોષ્ટક 11. ADMUX Register

7	6	5	4	3	2	1	0
REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0

Bit વર્ણન:

- **REFS1:0:** Reference Selection Bits.
 - 00: AREF, Internal Vref turned off.
 - 01: AREF pin પર external capacitor સાથે AVCC.
 - 11: Internal 2.56V Voltage Reference.
- **ADLAR:** ADC Left Adjust Result. જો 1 હોય, તો result left-adjusted છે (8-bit precision માટે).
- **MUX4:0:** Analog Channel અને Gain Selection Bits. કઈ ADC pin (ADC0-ADC7) connect કરવી તે select કરે છે.

મેમરી ટ્રીક

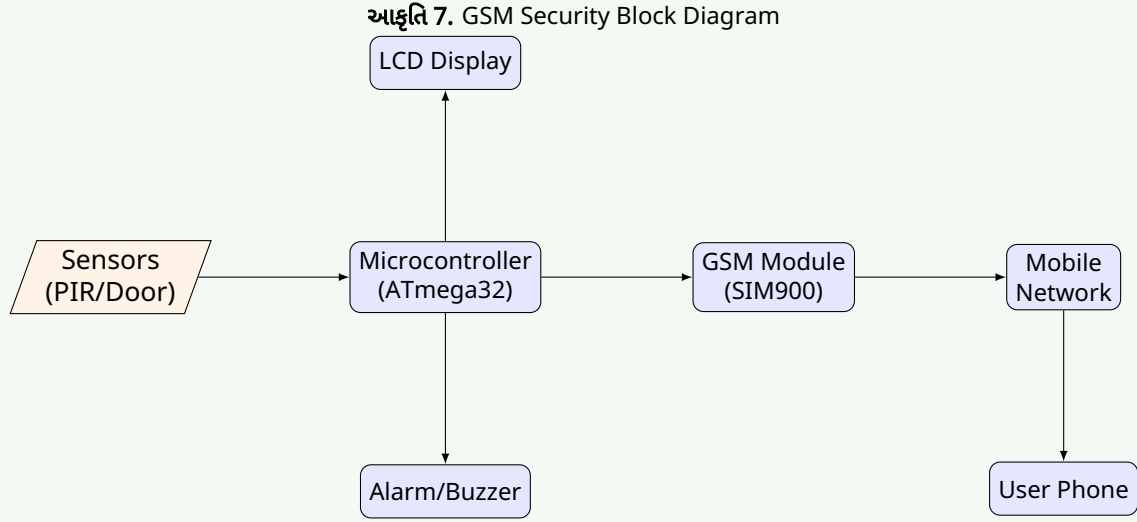
``Reference Adjust Multiplex Channel``

પ્રશ્ન 5(ક) [7 ગુણ]

GSM આધારિત સિક્યુરિટી સિસ્ટમનો બ્લોક ડાયાગ્રામ સમજાવો.

જવાબ

GSM Security System:



સ્ટેપ્સ:

- **Sensors:** PIR sensor motion detect કરે છે, Door switches entry detect કરે છે.
- **Microcontroller:** Sensor inputs process કરે છે. જો intrusion detect થાય:
 1. Alarm/Buzzer active કરે છે.
 2. GSM Module ને AT commands મોકલે છે.
 3. LCD પર Status update કરે છે.
- **GSM Module:** Cellular network સાથે UART દ્વારા communicate કરે છે. User ને SMS/Call મોકલે છે.
- **User Mobile:** Alert receive કરે છે ("Intruder Detected!").

મેમરી ટ્રીક

"Sensors Microcontroller GSM Mobile Alarm Display Power Operation"

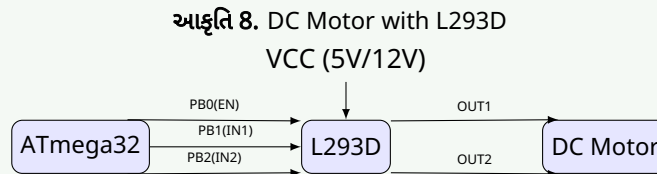
OR

પ્રશ્ન 5(અ) [3 ગુણ]

મોટર ડ્રાઇવર L293D નો ઉપયોગ કરી ડી.સી. મોટરને ATmega32 સાથે ઇન્ટરફેસ કરવાનો સર્કિટ ડાયાગ્રામ દોરો.

જવાબ

DC Motor Interface:



Control Logic:

- **Enable:** Driver channel activate કરવા High (1) આપો.
- **Direction:** IN1=1, IN2=0 (Forward); IN1=0, IN2=1 (Reverse).
- **Stop:** IN1=0, IN2=0.

મેમરી ટ્રીક

"Enable Direction Output Power Control"

OR

પ્રશ્ન 5(બ) [4 ગુણ]

ADCSRA રજિસ્ટર દોરો અને સમજાવો.

જવાબ

ADCSRA (ADC Control and Status Register A):

કોષ્ટક 12. ADCSRA Register

7	6	5	4	3	2	1	0
ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0

Bit કાર્યો:

- **ADEN:** ADC Enable. ADC વાપરવા માટે 1 set કરવું આવશ્યક છે.
- **ADSC:** Start Conversion. Single conversion શરૂ કરવા 1 write કરો.
- **ADATE:** Auto Trigger Enable.
- **ADIF:** Interrupt Flag. Conversion પૂરું થાય ત્યારે set થાય છે.
- **ADIE:** Interrupt Enable. જો Global Interrupts enabled હોય તો ADC interrupt activate કરે છે.
- **ADPS2:0:** Prescaler Select. System clock ને divide કરે છે (e.g., 8MHz crystal માટે /128) જેથી ADC clock (50-200kHz) મળે.

મેમરી ટ્રીક

"Enable Start Auto Interrupt Prescaler Configure"