

# Subject Name (Gujarati)

4351102 -- Summer 2025

Semester 1 Study Material

Detailed Solutions and Explanations

## પ્રશ્ન 1(અ) [3 માર્ક્સ]

ATmega32 ની લાક્ષણિકતાઓ જણાવો.

જવાબ

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

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

યાદી માટે: "Fast SRAM Enjoys Input Timers And Communication"

## પ્રશ્ન 1(બ) [4 માર્ક્સ]

માઈક્રોકંટ્રોલર પસંદ કરવા માટેના માપદંડો લખો.

જવાબ

માપદંડો	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

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

યાદી માટે: "Processing Memory I/O Power Cost Development Application Communication Package"

## પ્રશ્ન 1(ક) [7 માર્ક્સ]

એમ્બેડેડ સિસ્ટમનો સામાન્ય બ્લોક ડાયાગ્રામ દોરો અને સમજાવો.

## Mermaid Diagram (Code)

```

{Shaded}
{Highlighting} []
graph TD
    A[Input Devices] --{} B[Processor/Microcontroller]
    B --{} C[Output Devices]
    B --{} D[Memory]
    B --{} E[Communication Interface]
    F[Power Supply] --{} B
    G[Clock/Timer] --{} B
{Highlighting}
{Shaded}

```

## 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

યાદી માટે: "Processors Memory Input Output Communication Power Clock"

## પ્રશ્ન 1(ક OR) [7 માર્ક્સ]

રીયલ ટાઈમ ઓપરેટિંગ સિસ્ટમને વ્યાખ્યાયિત કરો અને તેની લાક્ષણિકતાઓ સમજાવો.

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

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

- **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

યાદી માટે: "Deterministic Preemptive Multitasking Fast Priority Interrupt"

## પ્રશ્ન 2(અ) [3 માર્ક્સ]

ATmega32 નો પીન ડાયાગ્રામ દોરો.

```

+{--}{--}U{--}{--}{--}+
PBO |1    40| PAO
PB1 |2    39| PA1

```

ચાલી માટે: ``Port B A Reset Vcc Ground Crystal Port D C''

### ATmega32 નો સ્ટેટસ રજિસ્ટર સમજાવો.

ચાલી માટે: "I Think Half Sign Overflow Negative Zero Carry"

## ATmega32 ની ડેટા મેમરી વિગતવાર સમજાવો.

```
{Shaded}
{Highlighting}[]
graph TD
    A[Data Memory Space {-}{-}{}] --> B[General Purpose Registers R0{-}R31]]
    A {-}{-}{-} --> C[I/O Memory 0x20{-}0x5F]]
```

```

A {-{-}{-} D[Extended I/O 0x60{-}0xFF]}
A {-{-}{-} E[Internal SRAM 0x100{-}0x8FF]}
{Highlighting}
{Shaded}

```

મેમરી સંગઠન:

- **General Purpose Registers:** Data operations માટે 32 registers (R0-R31)
- **I/O Memory:** Peripheral control registers નો direct access
- **Extended I/O:** Additional peripheral registers અને stack pointer
- **Internal SRAM:** Variables અને stack માટે 2KB volatile memory
- **Address Space:** 0x00 થી 0x8FF સુધી linear addressing
- **Stack Operation:** High memory addresses થી downward grows થાય છે

યાદી માટે: "General I/O Extended SRAM Address Stack"

## પ્રશ્ન 2(અ OR) [3 માર્ક્સ]

DDRx, PINx અને PORTx રજિસ્ટરના કાર્યો લખો.

જવાબ

Register	Function
DDRx	Data Direction Register - pin ને input/output તરીકે configure કરે છે
PINx	Pin Input Register - current pin state read કરે છે
PORTx	Port Output Register - output pins પર data write કરે છે

- **DDRx Bit:** 1 = Output, 0 = Input configuration
- **PINx Read:** Pins પર actual voltage level return કરે છે
- **PORTx Write:** Pin output હોય ત્યારે output state control કરે છે

યાદી માટે: "Direction Input Output"

## પ્રશ્ન 2(બ OR) [4 માર્ક્સ]

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

જવાબ

Register	Function
EEARH/EEARL	EEPROM Address Register (9-bit address)
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
- **EERE:** EEPROM Read Enable

**Programming Sequence:** Address set કરો → Dataset → Masterwriteenable → Writeenable

યાદી માટે: "Address Data Control Ready Master Write Read"

પ્રશ્ન 2(ક OR) [7 માર્ક્સ]

ક્લોક સોર્સને AVR સાથે જોડવાની વિવિધ રીતો સમજાવો.

જવાબ

Clock Source	વર્ણન
External Crystal	High precision, 1-16MHz
External RC	Low cost, moderate precision
Internal RC	Built-in 1/8MHz, કોઈ external components નહીં
External Clock	External clock signal input

Mermaid Diagram (Code)

```
{Shaded}
{Highlighting}[]
graph TD
    A[Crystal Oscillator] --{-}{-}{ D[AVR Microcontroller]}
    B[RC Oscillator] --{-}{-}{ D}
    C[Internal RC] --{-}{-}{ D}
    E[External Clock] --{-}{-}{ D}
{Highlighting}
{Shaded}
```

Clock Configuration:

- **Fuse Bits:** CKSEL3:0 અને SUT1:0 clock source select કરે છે
- **Startup Time:** Different sources માં અલગ અલગ startup delays હોય છે
- **Frequency Range:** Internal RC 1MHz અથવા 8MHz provide કરે છે
- **External Components:** Crystal ને stability માટે capacitors જોઈએ છે

યાદી માટે: "Crystal RC Internal External Fuse Startup Frequency Components"

પ્રશ્ન 3(અ) [3 માર્ક્સ]

Timer 1 સાથે સંકળાયેલા રજિસ્ટરોનું કાર્ય લખો.

જવાબ

Register	Function
TCNT1H/TCNT1L	Timer/Counter 1 register (16-bit)
TCCR1A/TCCR1B	Timer/Counter 1 Control registers
ICR1H/ICR1L	Input Capture register
OCR1AH/OCR1AL	Output Compare A register
OCR1BH/OCR1BL	Output Compare B register

- **TIMSK:** Timer Interrupt Mask register
- **TIFR:** Timer Interrupt Flag register

યાદી માટે: "Timer Control Input Output Mask Flag"

પ્રશ્ન 3(બ) [4 માર્ક્સ]

Timer 0 ને સામાન્ય મોડમાં પ્રોગ્રામ કરવાના મુદ્દાઓ લખો.

જવાબ

Programming Steps:

1. **Timer Mode Set કરો:** Normal mode માટે TCCR0 configure કરો

2. **Prescaler Select કરો:** Clock division ratio choose કરો
3. **Initial Value Load કરો:** TCNT0 register set કરો
4. **Interrupts Enable કરો:** જરૂર હોય તો TIMSK માં TOIE0 set કરો
5. **Timer Start કરો:** TCCR0 માં prescaler bits set કરો

```
TCCR0 = 0x05;    // Normal mode, prescaler 1024
TCNT0 = 0x00;    // Initial value
TIMSK |= 0x01;   // Enable overflow interrupt
```

યાદી માટે: "Set Select Load Enable Start"

### પ્રશ્ન 3(ક) [7 માર્ક્સ]

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

**જવાબ**

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

void USART\_Init() \{
    // Baud rate 9600 set    (8MHz clock  )
    UBRRH = 0x00;
    UBRRL = 51;

    // Receiver enable
    UCSRB = (1{ }RXEN);

    // Frame format set    : 8 data bits, 1 stop bit
    UCSRC = (1{ }URSEL)|(3{ }UCSZ0);
\}

unsigned char USART\_Receive() \{
    // Data receive    wait
    while(!(UCSRA \& (1{ }RXC)));
    return UDR;
\}

int main() \{
    DDRA = 0xFF;        // PORTA output
    USART\_Init();       // USART initialize

    while(1) \{
        PORTA = USART\_Receive(); // Receive    display
    \}
    return 0;
\}
```

યાદી માટે: "Initialize Receive Display Loop"

### પ્રશ્ન 3(અ OR) [3 માર્ક્સ]

AVR માં સીરીયલ કોમ્યુનિકેશન સાથે સંકળાયેલા રજિસ્ટરોના કાર્યો લખો.

#### જવાબ

Register	Function
UDR	USART Data Register
UCSRA	USART Control and Status Register A
UCSRB	USART Control and Status Register B
UCSRC	USART Control and Status Register C
UBRRH/UBRRL	USART Baud Rate Registers

મુખ્ય કાર્યો: Data transmission/reception, status monitoring, control configuration  
યાદી માટે: "Data Control Status Baud"

#### પ્રશ્ન 3(બ OR) [4 માર્ક્સ]

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

#### જવાબ

##### Programming Steps:

1. **Baud Rate Set કરો:** UBRRH/UBRRL registers configure કરો
2. **Transmitter Enable કરો:** UCSRB માં TXEN bit set કરો
3. **Frame Format Set કરો:** UCSRC માં data bits, stop bits configure કરો
4. **Empty Buffer માટે Wait કરો:** UCSRA માં UDRE flag check કરો
5. **Data Load કરો:** UDR register માં data write કરો

```
void USART_Transmit(unsigned char data) \{\n    while(!(UCSRA & (1<UDRE))); // Empty buffer    wait\n    UDR = data;                // Data send\n\}
```

યાદી માટે: "Baud Enable Format Wait Load"

#### પ્રશ્ન 3(ક OR) [7 માર્ક્સ]

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

```
\#include {avr/io.h}\n#include {avr/interrupt.h}\n\nvolatile unsigned int timer_count = 0;\n\nISR(TIMER1_OVF_vect) \{\n    timer_count++;\n    if(timer_count == 1) \{\n        PORTB ^= (1<4); // PORTB.4 toggle\n        timer_count = 0;\n        TCNT1 = 49911; // 2ms delay    reload\n    }\n\}\n\nint main() \{\n    DDRB |= (1<4); // PORTB.4 output\n\n    // Timer1 Normal mode, no prescaler
```

ગણતરી: 8MHz clock સાથે 2ms delay માટે:  $8\text{MHz} \times 2\text{ms} = 16000\text{cycles}$   
 $\text{Timer1counts} : 65536 - 16000 = 49536(\text{adjustment})$   
 યાદી માટે: "Configure Timer Calculate Enable Loop"





કનેક્શન વિગતો:

- VCC: +5V supply સાથે connect કરો
- OUT: ADC channel (PA0) પર analog output
- GND: Ground સાથે connect કરો
- Output:  $10\text{mV}/\text{linearvoltageoutput}$

યાદી માટે: "VCC OUT GND Linear"

#### પ્રશ્ન 4(બ OR) [4 માર્ક્સ]

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

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

int main() \{
    DDRC \&= {(1}{5});    // PC5 input
    DDRB = 0xFF;          // PORTB output
    PORTC |= (1}{5);      // PC5 pull-up enable }

    while(1) \{
        if(PINC \& (1}{5)) \{    // Bit 5 HIGH      check
            PORTB = 0x55;      // PORTB 55H send
        \}
        else \{
            PORTB = 0xAA;      // PORTB AAH send
        \}
    \}

    return 0;
\}
```

પ્રોગ્રામ Logic:

- **Bit Monitor:** Bit masking વાપરીને PC5 status check કરો
- **Conditional Output:** Input આધારે અલગ અલગ values send કરો
- **Continuous Loop:** Changes માટે continuously monitor કરો

યાદી માટે: "Monitor Conditional Output Loop"

#### પ્રશ્ન 4(ક OR) [7 માર્ક્સ]

AVR માં SPI ને પ્રોગ્રામ કરવા માટેના રજિસ્ટરોની ચર્ચા કરો.

જવાબ

Register	Function
SPCR	SPI Control Register
SPSR	SPI Status Register
SPDR	SPI Data Register

**SPCR Control Bits:**

- **SPIE:** SPI Interrupt Enable
- **SPE:** SPI Enable
- **DORD:** Data Order (MSB/LSB first)
- **MSTR:** Master/Slave Select
- **CPOL:** Clock Polarity
- **CPHA:** Clock Phase
- **SPR1:0:** SPI Clock Rate Select

**SPSR Status Bits:**

- **SPIF:** SPI Interrupt Flag
- **WCOL:** Write Collision Flag
- **SPI2X:** Double SPI Speed

**Programming Sequence:** SPCR configure કરો → *SPIenable* → *SPDRwrite* → *SPIFwait* → *SPDRread*  
 યાદી માટે: "Control Status Data Configure Enable Write Wait Read"

**પ્રશ્ન 5(અ) [3 માર્ક્સ]**

મોટર ડ્રાઇવર આઈસી L293D નો પીન ડાયાગ્રામ દોરો.

**જવાબ**

L293D  
 +{--}U{--}{--}+}  
 EN1 | 1    16 | VCC  
 1A | 2    15 | 4A  
 1Y | 3    14 | 4Y  
 GND | 4    13 | GND  
 GND | 5    12 | GND  
 2Y | 6    11 | 3Y  
 2A | 7    10 | 3A  
 VCC | 8    9 | EN2  
 +{--}{--}{--}{--}+}

**Pin કાર્યો:**

- **EN1, EN2:** Motor control માટે enable pins
- **1A, 2A, 3A, 4A:** Microcontroller થી input pins
- **1Y, 2Y, 3Y, 4Y:** Motors પર output pins
- **VCC:** Logic અને motor supply voltages
- **GND:** Ground connections

યાદી માટે: "Enable Input Output Supply Ground"

**પ્રશ્ન 5(બ) [4 માર્ક્સ]**

ADMUX રજિસ્ટર દોરો અને સમજાવો.

**જવાબ**

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





- **Humidity Sensor:** Moisture content monitor કરે છે (DHT22)
- **Pressure Sensor:** Atmospheric pressure changes detect કરે છે
- **Rain Sensor:** Precipitation levels detect કરે છે
- **Microcontroller:** Data collection માટે central processing unit
- **LCD Display:** Local visual data presentation
- **Data Logger:** Historical weather data store કરે છે
- **Wireless Module:** Remote locations પર data transmit કરે છે
- **Alarm System:** Extreme weather conditions માટે alerts
- **Power Supply:** System ને stable power પૂરું પાડે છે

**ઓપરેશન:** Sensors data collect કરે છે → *Microcontroller processes* → *Display updates* → *Datalogging* → *Wireless transmission* → *Alert generation*

**યાદી માટે:** "Temperature Humidity Pressure Rain Microcontroller Display Logger Wireless Alarm Operation"