**FMTH0301/Rev.5.3**

**Course Plan**

|  |  |
| --- | --- |
| Semester: IV | Year: 2022-23 |
| Course Title: Microcontroller: Programming and Interfacing | Course Code:22ECAC206 |
| Total Contact Hours: 15 hrs + 90 hrs | Credits: 1-0-3 |
| ISA Marks: 100 |  |
| Lesson Plan Author: Aruna S. Nayak, Namrata D. Hiremath, Umadevi F M | Date: 28.3.2023 |
| Checked By: Dr. S. R. Chickerur | Date: 31.3.2023 |

**Prerequisites:** Fundamental knowledge on Logic design concepts, Computer Organization

and Architecture and Programming

**Course Outcomes (COs):**

At the end of the course the student should be able to:

1. Identify the requirements of a real world problem and choose appropriate sensors/peripherals/ interfaces.
2. Reflect on RISC based programming in assembly and high level language.
3. DeconstructAVR architecture to interface with external environment.
4. Predict the key performance enhancement techniques in RISC based AVR microcontroller design.
5. Create solutions forindustry specified problems using microcontroller and associated tools.

**Course Articulation Matrix: Mapping of Course Outcomes (COs) with Program Outcomes (POs)**

|  |  |
| --- | --- |
| Course Title: **Microcontroller: Programming and Interfacing** | Semester: **IV** |
| Course Code: **21ECSC206** | Year: **2022-23** |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Course Outcomes (COs) / Program Outcomes (POs) | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| 1. Identify the requirements of a real world problem and choose appropriate sensors/peripherals/ interfaces. |  | H |  |  |  |  |  |  | M |  |  |  |  |  |
| 1. Reflect on RISC based programming in assembly and high level language. | H |  |  |  | H |  |  |  | M |  |  |  |  |  |
| 1. DeconstructAVR architecture to interface with external environment. |  |  |  | M | H |  |  |  | M |  |  |  |  |  |
| 1. Predict the key performance enhancement techniques in RISC based AVR microcontroller design.   . |  |  | H |  |  |  |  |  |  |  |  |  |  |  |
| 1. Createsolutions forindustry specified problems using microcontroller and associated tools. |  | H | M | M | L |  |  |  | M | M |  |  |  |  |

Degree of compliance **L**: Low **M**: Medium **H**: High

# Competency addressed in the Course and corresponding Performance Indicators

|  |  |
| --- | --- |
| **Competency** | **Performance Indicators** |
| **1.3:** Demonstrate competence in engineering fundamentals. | **1.3.1:** Apply programming concepts. |
| **1.4:** Demonstrate competence in computer science engineering knowledge. | **1.4.4:** Apply machine dependent/independent features to build system modules. |
| **2.1:** Demonstrate an ability to identify and characterize an engineering problem. | **2.1.2**:Identify processes, modules, variables, and  parameters of a computer based system to solve the problems. |
| **3.4:** Demonstrate an ability to advance an engineering design to defined  end state | **3.4.1:** Refine a conceptual design into a detailed design within the existing constraints (of the resources). |
| **4.1:** Demonstrate their ability to conduct investigations of technical issues  consistent with their level of knowledge and understanding | **4.1.2**: Relate modern engineering experimentation including experiment design, data acquisition, analysis and presentation |
| **5.3:** Demonstrate an ability to apply IT tools for the chosen engineering activity | **5.3.1:** Demonstrate proficiency in using IT tools for performing engineering activity |
| **9.3:** Demonstrate success in a team-based project | **9.3.1:** Present results as a team, with smooth integration of contributions from all individual efforts |
| **10.3:** Demonstrate the ability to integrate different modes of communication. | **10.3.1:** Create engineering-standard figures, reports and drawings to complement writing and presentations |

Eg: 1.2.3: Represents Program Outcome ‘1’, Competency ‘2’ and Performance Indicators ‘3’.

**Course Content**

|  |  |  |
| --- | --- | --- |
| **Module - I** | | |
| **Lecture /Reading** | **Introduction to Microcontroller and Embedded System**  Microcontrollers and General Purpose Microprocessors, Embedded System Features, Choosing a microcontroller, Criteria for choosing a microcontroller, Harvard and Von Neumann Architecture, Introduction to AVR Microcontroller and Arduino Family. | 01-hrs |
| **Hands on** | * Introduction to the hardware, setup, familiarizations with the working of the hardware | 03-hrs |
| **Lecture /Reading** | **AVR Architecture and Assembly Language Programming on AVR Microcontrollers**  Simplified View of an AVR Microcontroller, Internal Architecture (Harvard) of AVR, Registers and Data Memory in AVR, Instruction format and size in AVR, Using Instructions with Registers and Data Memory, Watch Dog Timer, Flags and Special Function Registers, Data Formats and Assembler directive.  Introduction to AVR Assembly Programming, Instruction Types and Instruction Set of AVR (Data Transfer Instructions, Branch Instructions, Bit and Bit test Instructions, Arithmetic and Logic Instructions, MCU Control Instructions, Jump and RET Instruction), Structure of Assembly Program in AVR, asm, lst, map and object files, Executing a program instruction by instruction, RISC Architecture features of AVR Microcontrollers, Viewing registers and memory with AVR Studio IDE. | 03-hrs |
| **Hand on** | * Assembly programming on the hardware using appropriate SDK   Set of programs to be given on various instruction types/ instruction set   * HLL Python programming on the hardware | 21-hrs |
| **Review** | Review I | 03-hrs |
| **Module – II** | | |
| **Lecture /Reading** | **AVR Time Delay**  Delay Calculation of AVR, AVR Multistage execution Pipeline, Timers/Counters, C Data Types | 02 hrs |
| **Hands on** | AVR Timer/Counter Programming | 06-hrs |
| **Lecture /Reading** | **AVR I/O Port Programming**  I/O Port Pins and their functions, Role of DDR/DDRx Registers in Input and output operations, Programming for I/O Ports,I/O Bit Manipulations, | 02-hrs |
| **Hands on** | I/O Port programming | 06-hrs |
| **Review** | Review II | 03-hrs |
|  | **Module – III** |  |
| **Lecture /Reading** | **Interrupts in AVR and Interrupt Programming**  AVR Interrupts, Interrupts vs Polling, Interrupt Service Routine, Steps in executing an interrupt, Sources of Interrupts, Interrupt Priority, Concept of Context Saving in task switching, Enabling and Disabling Interrupts, Programming Timer Interrupts, Programming external interrupts, | 02-hrs |
| **Hands on** | Interrupt Programming | 09-hrs |
| **Lecture /Reading** | **AVR Serial Port Programming**  Basics of Serial Communication, RS232 standards, RS232 Pins, RS232 Handshaking Signals, ATMEGA32 connections to RS232, Baud Rate and UBRR Register, UDR register and USART, UCSR Registers and USART Configuration, Programming AVR for Serial Communication. | 01-hrs |
| **Hands on** | Serial Communication programming | 06-hrs |
| **Review** | Review III | 03-hrs |
| **Module – IV** | | |
| **Lecture /Reading** | **LCD and Keyboard Interfacing**  LCD Interfacing, Sending Commands and Data to LCD (4 Bits and/or 8 Bits at a time). | 02 hrs |
| **Hands on** | Keyboard Interfacing, Matrix Keyboard connection to AVR Ports, Key Identification, | 06-hrs |
| **Lecture /Reading** | **Chapter No. 8. ADC, DAC and Sensor Interfacing**  Need for ADC and DAC in Interfacing, ADC Characteristics, ADC devices, and ATmega32 ADC features, Programming A/D Converter | 02 hrs |
| **Hands on** | DAC Interfacing, Sensor Interfacing | 03-hrs |
| **Review** | Review IV | 03-hrs |
| **Module – V** | | |
| **Hands on** | **Integration of the work done in various modules according to the problem statement** | 12-hrs |
| **Review** | Review V | 03-hrs |
| **Final Evaluation** | Review VI: Presentation + Project exhibition | 03-hrs |
| **Text Books:**   1. Mazidi M. A, Naimi Sarmad, Naimi Sepehr, “”The AVR Microcontroller and Embedded System using Assembly and C”, Prentice Hall.0 | | |
| **Reference Books:**   1. J. M. Hughes, “Arduino A Technical Reference”, O’Reilly | | |

**Evaluation scheme**

|  |  |  |
| --- | --- | --- |
| **Sl.No.** | **Assessment** | **Marks** |
|  | Module 1 Hands –on | **15** |
|  | Module 2 Hands –on | **15** |
|  | Module 3 Hands –on | **20** |
|  | Module 4 Hands –on | **20** |
|  | Module 5 Hands –on | **20** |
|  | Final review Project Demonstration/ Exhibition/ Competition | **10** |

|  |  |
| --- | --- |
| **Date:** | **Head, SoCSE** |

**Course Assessment Plan**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Course Title: Microcontroller: Programming & Interfacing Code: 21ECSC206 | | | | | | | | | |
| Course outcomes (COs) | Weightage in assessment | Review-1 | Review-2 | | Review-3 | | Review-4 | Review-5 | Final Review |
| 1. **Identify** the requirements of a real world problem and choose appropriate sensors/peripherals/ interfaces. | 25% | √ | | -- | √ | -- | | -- | -- |
| 1. **Reflect** on RISC based programming in assembly and high level language. | 15% | -- | | √ | √ | -- | | -- | -- |
| 1. **Deconstruct** AVR architecture to interface with external environment. | 20% | -- | | -- | √ | √ | | -- | -- |
| 1. **Predict** the key performance enhancement techniques in RISC based AVR microcontroller design.   . | 10% | -- | | -- | √ | -- | | -- | -- |
| 1. **Create** solutions forindustry specified problems using microcontroller and associated tools. | 30% | -- | | -- | -- | -- | | √ | √ |
| Weightage |  | 15% | | 15% | 20% | 20% | | 20% | 10% |

**Assessment Rubrics**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Criteria** | **Good** | **Satisfactory** | **Scope for improvement** |
| **Module-1** | **Identify the components and specification as per requirements**  **(15M)** | * Correctly identified the required specifications of hardware components. * Correctly identified all the required hardware components**.(12-15M)** | * Partially identified the required specifications of hardware components. * Partially identified all the required hardware components**.(6-11M)** | Incorrectly identified the required specifications of hardware components**. (0-5M)** |
| **Module-2** | **Analyze and implement module/program**  **(15M)** | Applies instructions, program constructs, procedures, principles, and themes appropriately in new contexts.  **(12-15M)** | Applies instructions, program constructs, procedures, principles, and themes with minor inaccuracies. **(6-11M)** | Inaccurately and inappropriately applies instructions, program constructs, procedures, principles, and themes. **(0-5M)** |
| **Module-3** | **Identify the scenario/situations which require interrupts**  **(10M)** | Able to correctly Identify the scenario/situations which require interrupts  **(8-10M)** | Able to partially Identify the scenario/situations which require interrupts  **(3-7M)** | Unable to Identify the scenario/situations which require interrupts.  **(0-2M)** |
| **Justify the choice of interrupt with implementation in the project (10M)** | Able to correctly justify the choice of interrupt with correct implementation in the project **(8-10M)** | Able to correctly justify the choice of interrupt with incorrect implementation in the project  **(3-7M)** | Unable to justify the choice of interrupt.  **(0-2M)** |
| **Module-4** | **Interface peripherals for UI interactions**  **(10M)** | Able to correctly Interface peripherals for UI interactions.  **(8-10M)** | Able to partially Interface peripherals for UI interactions. **(3-7M)** | Unable to interface peripherals. **(0-2M)** |
| **Interface appropriate sensors for the solution model**  **(10M)** | Able to correctly Interface appropriate sensors for the solution model.  **(8-10M)** | Able to partially Interface appropriate sensors for the solution model..  **(3-7M)** | Unable to interface appropriate sensors. **(0-2M)** |
| **Module-5** | **Integration of the developed modules,**  **(20M)** | Able to correctly perform functional testing and final prototype development.  **(15-20M)** | Able to partially perform functional testing and final prototype development.  **(8-14M)** | Unable to perform functional testing and final prototype development **(0-7M)** |
| **Final** | **Project Demonstration, Presentation (10M)** | Successfully able to exhibit and participate in project competition **(8-10M)** | Able to exhibit and not able to participate in project competition  **(3-7M)** | Unable to exhibit or participate in project competition. **(0-2M)** |

**Module wise Plan**

|  |  |
| --- | --- |
| Course Code and Title: Microcontroller: Programming & Interfacing (21ECSC206) | |
| Module Number and Title: **1. Introduction to Microcontroller, Embedded System, AVR Architecture and Assembly Language Programming on AVR Microcontrollers** | Planned Hours: 4 hrs (teaching) + 24 hrs (hands-on) +3 hrs (Review) |

**Learning Outcomes:-**

**At the end of the topic the student should be able to:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Topic Learning Outcomes** | **COs** | **BL** | **PI Code** |
| 1. Identify the components and specification as per requirements. | CO1 | L4 | 2.1.2 |
| 1. Write assembly and high level language programs based on RISC architecture. | CO2 | L4 | 1.3.1  5.3.1 |
| 1. Analyse the flow of data between microcontroller & respective components for project. | CO3 | L4 | 4.1.2  5.3.1  9.3.1 |
| 1. Present results as a team for the assigned task | CO1 | L4 | 9.3.1 |

|  |
| --- |
| Lesson Schedule  Class No. - Portion covered per hour |
| 1. Microprocessors vs. Microcontrollers, AVR Types, AVR internal architecture, AVR’s CPU, Data Address Space, AVR Ports |
| 1. Introduction to the hardware, setup, |
| 1. Familiarizations with the working of the hardware |
| 1. Working on Data transfer instructions using AVR Studio |
| 1. Addressing modes |
| 1. Arithmetic & Logical instructions |
| 1. Unconditional Branch instructions |
| 1. Introduction to control flow instruction |
| 1. Conditional branch instruction |
| 1. Loop instruction |
| 1. Assembly programs on simulator |
| 1. Assembly programs on simulator…(contd.) |
| 1. Assembly programming on the hardware using appropriate SDK, |
| 1. Set of programs to be given on various instruction types/ instruction set |
| 1. HLL Python programming on the hardware |

**Review Questions**

|  |  |  |  |
| --- | --- | --- | --- |
| Sl.No. - Questions | TLOs | BL | PI Code |
| 1. Identify the components required to build atemperature controlled DC fan and list their specifications. | 1 | L4 | 2.1.2 |
| 1. Write an assembly code that allocates an array with the size of 2 bytes on the stack. | 2 | L4 | 1.3.1  5.3.1 |
| 1. Develop and present a project to demonstrate the display of a key press on the LCD screen. | 3 | L4 | 4.1.2  9.3.1 |

|  |  |
| --- | --- |
| Course Code and Title: Microcontroller: Programming & Interfacing (21ECSC206) | |
| Module Number and Title: **2. AVR Time Delay and I/O Port Programming.** | Planned Hours: 4 hrs (teaching) + 12 hrs (hands-on) +3 hrs (Review) |

**Learning Outcomes:-**

**At the end of the topic the student should be able to:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Topic Learning Outcomes** | **COs** | **BL** | **CA Code** |
| 1. Analyse and implement module/program using I/O ports | CO3 | L4 | 4.1.2  5.3.1  9.3.1 |
| 1. Analyse and implement module/program using the, Timers & Counters | CO3 | L4 | 4.1.2  5.3.1  9.3.1 |

|  |
| --- |
| Lesson Schedule  Class No. - Portion covered per hour |
| **AVR Time Delay and Instruction Pipeline**   1. Delay Calculation of AVR, AVR Multistage execution Pipeline, Timers/Counters, C Data Types |
| 1. Timer features |
| 1. Timer programming modes |
| 1. Assembly programs on Timer usage |
| 1. Counter features |
| 1. Counter programming modes |
| 1. Assembly programs on Counter usage |
| 1. I/O Port Programming |
| 1. I/O Port Pins and their functions |
| 1. Role of DDR/ DDRx Registers in Input and output operations |
| 1. Programming for I/O Ports |
| 1. I/O Bit Manipulations |
| 1. Assembly programs on I/O Ports |
| 1. Assembly programs on I/O Ports…(contd.) |

**Review Questions**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sl.No. - Questions | TLOs | | BL | PI Code |
| 1. It is required to measure the moisture content of soil in a field. How do you setup the equipment to read the required data? Discuss the specifications of the sensors to be used and how to make the data ready to be read into the Ports of the microcontroller. | TLO1 | | L4 | 4.1.2  5.3.1  9.3.1 |
| 1. Propose a system based on AVR microcontroller that evaluates the traffic density using suitable sensors and accomplishes dynamic timing slots with different levels. | TLO 2 | | L4 | 4.1.2  5.3.1  9.3.1 |
| Course Code and Title: Microcontroller: Programming & Interfacing (21ECSC206) | | | | |
| Module Number and Title: **3. AVR Interrupt and Serial Port Programming** | | Planned Hours: 3 hrs (teaching) + 15 hrs (hands-on) +3 hrs (Review) | | |

**Learning Outcomes:-**

**At the end of the topic the student should be able to:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Topic Learning Outcomes** | **COs** | **BL** | **CA Code** |
| 1. Identify the scenario/situations which require interrupts. | CO 4 | L5 | 3.4.1 |
| 1. Justify the choice of interrupt with implementation in the project | CO 4 | L5 | 3.4.1 |

|  |
| --- |
| Lesson Schedule  Class No. - Portion covered per hour |
| 1. AVR Interrupts, Interrupts vs Polling, Steps in executing an interrupt, Concept of Context Saving in task switching, Enabling and Disabling Interrupts |
| 1. Interrupt Service Routine |
| 1. Sources of Interrupts |
| 1. Interrupt Priority |
| 1. Programming Timer Interrupts |
| 1. Programming Counter interrupts |
| 1. Programming external interrupts |
| 1. Basics of Serial Communication, RS232 standards, Baud Rate and UBRR Register, UDR register and USART, UCSR Registers and USART Configuration, |
| 1. RS232 Pins, RS232 Handshaking Signals |
| 1. ATMEGA32 connections to RS232 |
| 1. Programming AVR for Serial Communication. |
| 1. Programming AVR for Serial Communication…(contd.) |
| 1. Programming AVR for Serial Communication using interrupts |
| 1. Programming AVR for Serial Communication using interrupts…(contd.) |

**Review Questions**

|  |  |  |  |
| --- | --- | --- | --- |
| Sl.No. - Questions | TLOs | BL | PI Code |
| 1. Design a digital clock which functions as a stop watch, event counter and an alarm clock. Highlight the use of interrupts in this design. | TLO1 | L5 | 3.4.1 |
| 1. Consider a scenario of industrial environment which has hazardous working environment that include boilers, poisonous gases and hot liquid flowing through pipe. Various sensors are mounted in the vicinity to detect fire break out, liquid and gas leakages. The sensors are the integral part of safety system. The sensors provide inputs to the microcontroller based safety system. Justify the choice of interrupt and their priority for such use case. | TLO2 | L5 | 3.4.1 |

|  |  |
| --- | --- |
| Course Code and Title: Microcontroller: Programming & Interfacing (21ECSC206) | |
| Module Number and Title: **4. Interfacing LCD, Keyboard, ADC, DAC and Sensor to the microcontroller** | Planned Hours: 4 hrs (teaching) + 9 hrs (hands-on) +3 hrs (Review) |

**Learning Outcomes:-**

**At the end of the topic the student should be able to:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Topic Learning Outcomes** | **COs** | **BL** | **CA Code** |
| 1. Interface keypad and LCD display for UI interactions. | CO3 | L5 | 4.1.2  5.3.1  9.3.1 |
| 1. Interface appropriate sensors for the solution model | CO3 | L5 | 4.1.2  5.3.1  9.3.1 |

|  |
| --- |
| Lesson Schedule  Class No. - Portion covered per hour |
| 1. LCD Basics |
| 1. LCD Interfacing |
| 1. Sending Commands and Data to LCD (4 Bits and/or 8 Bits at a time). |
| 1. Sending Commands and Data to LCD (4 Bits and/or 8 Bits at a time)…contd. |
| 1. Keyboard basics |
| 1. Keyboard Interfacing |
| 1. Matrix Keyboard connection to AVR Ports |
| 1. Key Identification |
| 1. Basics on Data conversion |
| 1. Need for ADC and DAC in Interfacing |
| 1. ADC Characteristics |
| 1. ADC devices, and ATmega32 ADC features, Programming A/D Converter. |
| 1. DAC Interfacing |
| 1. Sensor Interfacing |

**Review Questions**

|  |  |  |  |
| --- | --- | --- | --- |
| Sl.No. - Questions | TLOs | BL | PI Code |
| 1. Design a digital calculator, which performs arithmetic functions on 8-bit data | TLO1 | L5 | 4.1.2 |
| 1. Design temperature controlled fan. | TLO2 | L5 | 4.1.2 |

|  |  |
| --- | --- |
| Course Code and Title: Microcontroller: Programming & Interfacing (21ECSC206) | |
| Module Number and Title: **5. Integration of the work done in various modules according to the problem statement** | Planned Hours: 12 hrs (hands-on) +6 hrs (Review) |

**Learning Outcomes:-**

**At the end of the topic the student should be able to:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Topic Learning Outcomes** | **COs** | **BL** | **CA Code** |
| 1. Integration of the developed modules, Functional Testing, Final prototype developments | CO5 | L5 | 1.3.1  2.1.2  3.4.1  5.3.1  9.3.1  10.3.1 |

|  |
| --- |
| Lesson Schedule  Class No. - Portion covered per hour |
| 1. Integration of the developed modules |
| 1. Integration of the developed modules |
| 1. Integration of the developed modules |
| 1. Functional Testing |
| 1. Functional Testing |
| 1. Functional Testing |
| 1. Final prototype developments |
| 1. Final prototype developments |
| 1. Final prototype developments |