# A-Level Computer Science

## Monitoring and Control Systems – Study Booklet

Prepared for Student Revision & Exam Preparation

# Table of Contents

# A-level Answer Sheet - 28A Monitoring and control systems

Monitoring and control system
A-Level
Activities
1.
Activity-1 Duration: 15 minutes
Design a microprocessor-controlled greenhouse. Include a flowchart and specify the sensors and actuators you will use.
Sensors to measure: Temperature Humidity Light intensity Gas (monitor O2 and CO2 levels) Motion (to detect rain) Actuators: Valves controlling sprinklers Heater to be turned ON when the temperature is low A fan to be turned ON when the temperature is high Motor to control the window screen (for light) Motor to control the roof (in case of rain)
End of topic questions
2.
End of topic questions
The optimum temperature for yeast production is 32■C - 35■C. Write a program to control the temperature of the manufacturing process that uses a cooler. SystemOFF = False While (SystemOFF = False) call readTemp (value) difference = 33.5 - value if (abs(difference) < 1.5) difference = 0 print (“System is under control”) else if (difference > 1.5) call setCoolerTemp (1.5, ‘L’) else if (difference < 1.5) call setCoolerTemp (1.5, ‘H’) end if //introducing time delay for i = 1 to 999999 end for end while A home security system monitors the motion detectors at several places and switches ON an alarm in case intrusion is detected. This system is controlled by a microprocessor. The bit 0 of location 38FE contains the status of the motion detector 1. Write an assembly language code to test this bit.
End of topic questions
Modify the code in question 2 to set the alarm ON. The alarm is controlled by location 2991. When bit 0 in 38FE is set to 1, bit 0 in location 2991 is set to 1 to switch ON the alarm.

# A-level Presentation - 28A Monitoring and control systems

Monitoring and control systems
A-Level
Lesson Objectives
Students will learn about: Sensors Actuators Programming in real-time Assembly language programming for testing sensors and controlling actuators
‹#›
Content
1.
Introduction
Monitoring and control systems are used widely in several areas such as home appliances, security systems, traffic lights, and industries. The basic approach is to measure a physical quantity such as temperature, pressure, light intensity, or volume of a gas. The measured value is sent to a computer. This is done by a sensor. A sensor is a hardware device that measures a physical quantity and transmits the value to a computer that controls the system.
‹#›
Introduction
There are two types of functions of a computer in these applications: Monitoring: The value received is checked with the allowable values. If the values do not satisfy the safety requirements, a warning is sent so that humans can take necessary action. The warning may be a text message or ringing of an alarm.
‹#›
Introduction
Monitoring and control: In addition to monitoring, some systems are designed to control the system. If a value is not within the allowable limits, the computer sends instructions to that part of the system that could reverse the cause of change. The part of the system that is responsible for control operation is the actuator. The actuator is a hardware device that receives the signals from the computer and adjusts the setting of a controlling device. In this system, there is feedback to continually process the measurements and take necessary action.
‹#›
A control system
‹#›
Analogue to Digital converter (ADC)
‹#›
Physical quantities are analogue in nature; that is, the values are constantly changing, and to measure these values, we need to interpret the readings. For example, to read the time using an analogue clock, we need to look at the hands of the clock to state the hours and minutes. Computers cannot read analogue values directly. Analogue values are converted to discrete digital values so that the computer can process them.
Sensors
‹#›
Some examples of sensors with their applications are given. Light: Switching on lights automatically at night and off during the day, control light intensity in greenhouse Temperature: monitor or control a chemical process; the temperature in a greenhouse pH: monitoring pollution in land and water resources Motion/ infrared: security systems
Digital to Analogue converter (DAC)
‹#›
The control signals from the computer are digital in nature and are converted to analogue signals. For example, a motor requires an electric signal instead of a digital signal from a computer.

Closed-loop feedback control system
‹#›
The feedback signal directly controls the operation. A microprocessor compares the desired value with the actual value received. Based on the difference, the processor sends a signal to the actuator.
Real-time programming
‹#›
In order to design monitoring or monitoring and control system, real-time programming is required. The values from the sensor are taken at specified time intervals and analyzed. Based on the application, the time intervals at which reading is taken shall be varied.
Monitoring: Example program
‹#›
The system is monitored all the time it is switched on. Based on the reading, the alarm is turned on through a procedure call. The alarm shall be designed to light up in different colours based on whether alarmInput = ‘H’ or ‘L’. Time delay is introduced using for loop. A boolean variable is also reset to avoid calling the warning procedure again within a time interval.
Controlling: Example program
‹#›
Similar to a monitoring application, a control application is also designed. The difference between the actual reading and the desired output value is calculated. Based on this difference, the control signal is sent to the actuator.
‹#›
The logic operations are: NOT: Complement’s the binary value AND: Produces output ‘1’ only when both the inputs are ‘1.’ OR: Produces output ‘1’ when at least one of the input is ‘1.’ XOR: Produces output ‘1’ when both the outputs are different, otherwise produces ‘0’. This is used for toggle operation.
Bitwise operations
‹#›
Bitwise operations
‹#›
These logical operations are used to manipulate bits in a number. For example: An OR function is used to convert some bits to ‘1’ without affecting the other bits. Similarly, an AND function is used to convert some bits to ‘0’ without affecting the other bits. XOR function is used to invert selected bits. This concept of manipulating bits, that is, setting selected bits true or false, is called masking.
Bitwise operations
‹#›
Microprocessors are used to control actuators. Based on the conditions, flag values can be set in a register. In some applications, there are more than one parameters to be controlled.
Bitwise operations
SET 0 if Difference1 < 0 and SET 1 if Difference1 > 0
SET 0 if Difference2 < 0 and SET 1 if Difference2 > 0
‹#›
Let us consider a situation in a control application, and the bit 0 in location 923F is be set to 1, and all other bits are to be set to 0. Assembly code is used to control bit 0 directly. B denotes a binary number, and to use a hexadecimal number, the symbol & is used. For example, &A3.;
Bitwise operations
‹#›
Assembly language instructions are also used to test whether the values from the sensors are within acceptable limits or not. Assume an application has 8 different sensors and each bit in a location denotes the status of a sensor. CMP is used for comparison, and JPN is used after a comparison instruction in order to jump to the location specified if the comparison is false. Alternatively, instruction JPE is used to jump to the location specified if the comparison is true.
Bitwise operations
‹#›
To check the status of sensor 2, this code is used:

Bitwise operations
‹#›
It is important to note that AND #operand means AND operation is performed between contents of accumulator and operand specified. The result of the operation is also stored in the accumulator. AND performs AND operation between contents of accumulator and contents in the address specified. Again, the result is stored in the accumulator.
Assembly language instructions
‹#›
The different load instructions are given in the table.
‹#›
Based on different addressing methods, there are different compare instructions too.
Let’s review some concepts
Real-time programming Monitoring: check the values from the sensor continuously by introducing time delay in the loop. A warning is sent in case the value is not within the allowable limits. Monitoring & control: In addition to monitoring, a control signal is set to the actuator by the difference between the actual reading and the desired output value
Bitwise operations NOT, OR, AND & XOR Masking: OR: set selected bits to ‘1’ AND: set selected bits to ‘0’ XOR: invert selected bits
Different types of load instructions Immediate: LDM #n, LDR #n Direct: LDD Indirect: LDI Indexed: LDX < address>
‹#›
Sensors & ADC A sensor is a hardware device that measures a physical quantity and transmits the value to a computer that controls the system. Analogue values are converted to discrete digital values by ADC so that the computer can process them.
Actuators & DAC The actuator is a hardware device that receives the signals from the computer and adjusts the setting of a controlling device. The control signals from the computer are digital in nature and are converted to analogue signals by DAC
Different types of compare instructions Immediate: CMP #n Direct: CMP Indirect: CMI
Activities
2.
Activity-1 Duration: 15 minutes
Design a microprocessor-controlled greenhouse. Include a flowchart and specify the sensors and actuators you will use.
‹#›
End of topic questions
3.
End of topic questions
The optimum temperature for yeast production is 32■C - 35■C. Write a program to control the temperature of the manufacturing process that uses a cooler. A home security system monitors the motion detectors at several places and switches ON an alarm in case intrusion is detected. This system is controlled by a microprocessor. The bit 0 of location 38FE contains the status of the motion detector 1. Write an assembly language code to test this bit. Modify the code in question 2 to set the alarm ON. The alarm is controlled by location 2991. When bit 0 in 38FE is set to 1, bit 0 in location 2991 is set to 1 to switch ON the alarm.
‹#›

# A-level Quiz - 28A Monitoring and control systems

Monitoring and control systems

A-Level

What bit manipulation operation is used to convert some bits in a register to 1 without affecting other values? NOT AND OR XOR What bit manipulation operation is used to convert some bits in a register to 0 without affecting other values? NOT AND OR XOR What bit manipulation operation is used to invert selected bits in a register? NOT AND OR XOR

Quiz

If the value at location 65F9 is B00100110, what is the value in the accumulator when the following codes are executed? LDM #B01001011 AND 65F9 01001011 00000010 01000000 00100110 What is the value in the accumulator when the following codes are executed? LDM #B11011011 OR #B00111100 11111111 00111100 11011011 00000000 If the value at location 3234 is B11001101, what is the value in the accumulator when the following codes are executed? LDD 3234 XOR #B10101010 01100111 11101111 10001000 10101010

If the value at location 65F9 is &3E, what is the value in the accumulator when the following codes are executed? LDM #&0F AND 65F9 00001111 00111110 00001110 00001101 In a control application, the actuator is controlled based on the location of 1329 in the microprocessor. What is the value in location 1329 after the execution of this code? 00000000 00100000 11011111 None of the above

In a monitoring application, the status of the sensor is stored at location 120F in a microprocessor. If the value at this location is 00000010, what happens when the given code is executed? In a monitoring application, the status of sensor is stored at location 120F in microprocessor. If the value at this location is 01000100, what happens when the given code is executed?

The code at the location labelled ‘process’ executes Value at 120F = #B00000000 Accumulator = #B00000000 Both B and C

The code at the location labelled ‘process’ executes Value at 120F = #B00000000 Accumulator = #B00000000 None of the above

Answers

# A-level Revision Notes - 28A Monitoring and control systems

Monitoring and control systems
A-Level
Revision notes
1.
Introduction
Monitoring and control systems are used widely in several areas such as home appliances, security systems, traffic lights, and industries. The basic approach is to measure a physical quantity such as temperature, pressure, light intensity, or volume of a gas. The measured value is sent to the computer. This is done by a sensor. A sensor is a hardware device that measures a physical quantity and transmits the value to a computer that controls the system. There are two types of functions of a computer in these applications: Monitoring: The value received is checked with the allowable values. If the values do not satisfy the safety requirements, a warning is sent so that humans can take necessary action. The warning may be a text message or ringing of an alarm. Monitoring and control: In addition to monitoring, some systems are designed to control the system. If a value is not within the allowable limits, the computer sends instructions to that part of the system that could reverse the cause of change. The part of the system that is responsible for control operation is the actuator. The actuator is a hardware device that receives the signals from the computer and adjusts the setting of a controlling device. In this system, there is feedback to continually process the measurements and take necessary action.
Figure 1: A control system
Analogue to Digital converter (ADC): Physical quantities are analogue in nature; that is, the values are constantly changing, and to measure these values, we need to interpret the readings. For example, to read the time using an analogue clock, we need to look at the hands of the clock to state the hours and minutes. Computers cannot read analogue values directly. Analogue values are converted to discrete digital values so that the computer can process them. Sensor: Some examples of sensors with their applications are given. Light: Switching on lights automatically at night and off during the day, control light intensity in greenhouse Temperature: monitor or control a chemical process; the temperature in a greenhouse pH: monitoring pollution in land and water resources Motion/ infrared: security systems Digital to Analogue converter (DAC): The control signals from the computer are digital in nature and are converted to analogue signals. For example, a motor requires an electric signal instead of a digital signal from a computer. A closed-loop feedback control system is given in the figure below. It can be noted that the feedback signal directly controls the operation. A microprocessor compares the desired value with the actual value received. Based on the difference, the processor sends a signal to the actuator.
Real time programming
In order to design monitoring or monitoring and control system, real-time programming is required. The values from the sensor are taken at specified time intervals and analyzed. Based on the application, the time intervals at which reading is taken shall be varied. SystemOFF = False ReadingOutOfRange = False While (SystemOFF = False) call readSensor (value) if (value > valueAllowed) ReadingOutOfRange = True print (“Reading is higher than allowable value”) alarmInput = ‘H’ else if (value < valueAllowed) ReadingOutOfRange = True print (“Reading is lower than allowable value”) alaramInput = ‘L’ end if If ReadingOutOfRange CALL alarmOn (alarmInput) end if ReadingOutOfRange = False //introducing time delay for i = 1 to 999999 end for end while In the above program, the system is monitored all the time it is switched on. Based on the reading, the alarm is turned on through a procedure call. The alarm shall be designed to light up in different colours based on whether alarmInput = ‘H’ or ‘L’. Time delay is introduced using for loop. A boolean variable is also reset to avoid calling the warning procedure again within a time interval.

Similar to a monitoring application, a control application is also designed. The difference between the actual reading and the desired output value is calculated. Based on this difference, the control signal is sent to the actuator. SystemOFF = False INPUT desiredOutputValue While (SystemOFF = False) call readSensor (value) difference = desiredOutputValue - value if (abs(difference) < desiredOutputValue/100) difference = 0 print (“System is under control”) else if (difference > 0) actuatorInputValue = difference/desiredOutputValue actuatorInputDirection = ‘L’ call actuator (actuatorInputValue, actuatorInputDirection) else if (difference < 0) actuatorInputValue = abs(difference)/desiredOutputValue actuatorInputDirection = ‘H’ call actuator (actuatorInputValue, actuatorInputDirection) end if //introducing time delay for i = 1 to 999999 end for end while
Bitwise operations
The logic operations are: NOT: Complement’s the binary value AND: Produces output ‘1’ only when both the inputs are ‘1.’ OR: Produces output ‘1’ when at least one of the input is ‘1.’ XOR: Produces output ‘1’ when both the outputs are different, otherwise produces ‘0’. This is used for toggle operation. An example of these operations is given in the figure below. These logical operations are used to manipulate bits in a number. For example: An OR function is used to convert some bits to ‘1’ without affecting the other bits. Similarly, an AND function is used to convert some bits to ‘0’ without affecting the other bits. XOR function is used to invert selected bits. This concept of manipulating bits, that is, setting selected bits true or false is called masking.
Microprocessors are used to control actuators. Based on the conditions, flag values can be set in a register. In some applications, there are more than one parameters to be controlled. Let us consider a situation in a control application and the bit 0 in location 923F is be set to 1 and all other bits are to be set to 0. Assembly code is used to control bit 0 directly. B denotes a binary number and to use a hexadecimal number, the symbol & is used. For example, &A3.; Assembly language instructions are also used to test whether the values from the sensors are within acceptable limits or not. Assume an application has 8 different sensors and each bit in a location denotes the status of a sensor. CMP is used for comparison, and JPN is used after a comparison instruction in order to jump to the location specified if the comparison is false. Alternatively, instruction JPE is used to jump to the location specified if the comparison is true.
SET 0 if Difference1 < 0 and SET 1 if Difference1 > 0
SET 0 if Difference2 < 0 and SET 1 if Difference2 > 0
To check the status of sensor 2, this code is used: It is important to note that AND #operand means AND operation is performed between contents of accumulator and operand specified. The result of the operation is also stored in the accumulator. AND performs AND operation between contents of accumulator and contents in the address specified. Again, the result is stored in the accumulator. The different load instructions are given in the table.
Based on different addressing methods, there are different compare instructions too.
Activities
2.
Activity-1 Duration: 15 minutes
Design a microprocessor-controlled greenhouse. Include a flowchart and specify the sensors and actuators you will use.
End of topic questions
2.
End of topic questions
The optimum temperature for yeast production is 32■C - 35■C. Write a program to control the temperature of the manufacturing process that uses a cooler. A home security system monitors the motion detectors at several places and switches ON an alarm in case intrusion is detected. This system is controlled by a microprocessor. The bit 0 of location 38FE contains the status of the motion detector 1. Write an assembly language code to test this bit. Modify the code in question 2 to set the alarm ON. The alarm is controlled by location 2991. When bit 0 in 38FE is set to 1, bit 0 in location 2991 is set to 1 to switch ON the alarm.