İndirilme Tarihi

05.02.2026 14:22:10

KNT238 - MICROCONTROLLER APPLICATIONS - Meslek Yüksekokulu - Elektronik ve Otomasyon Bölümü
General Info

Objectives of the Course

To teach the historical development process of microprocessors and microcontrollers. To understand the functions and operation of microcontroller hardware units. To write the appropriate program to control the microcontroller. To make button, LED, 7 segment display, matrix display, character LCD, keypad, ADC and serial communication applications with microcontroller.

Course Contents

Differences between microprocessor and microcontroller systems systems, microcontroller systems, programmer card, the program into machine language translation, the compiled program to microcontroller installation, algorithms, flow charts, the memory map of the microcontroller, the microcontroller commands editor program the microcontroller, the microcontroller basic blocks of the program, the basic input-output programs , microcontroller program, compile, run the compiled program step by step, push-button and LED applications with microcontroller, microcontroller circuit applications with a 7-segment display, keypad with microcontroller applications, lcd with microcontroller applications.

Recommended or Required Reading

Çiçek, S. (2021). CCS C ile PIC Programlama. Altaş Yayıncılık.<br /><br />Akademi B. (2014). Herkes için PIC Programlama. Pusula Yayıncılık, Documents prepared by the relevant lecturer / Computer and Projector

Planned Learning Activities and Teaching Methods

Lecturing, application, question and answer, group work, skills development training

Recommended Optional Programme Components

Students must follow laboratory rules.

Instructor's Assistants

....

Presentation Of Course

The course is taught in a computer lab using microcontroller training kits.

Dersi Veren Öğretim Elemanları

Inst. Samet Ayık

Program Outcomes

1. It can learn the differences between microprocessors and microcontrollers.
2. It can perform operations with the hardware units and input/output units of the microcontroller.
3. It can learn high-level and low-level languages and the differences between them.
4. The C programming language can be converted into machine language using a compiler program and then uploaded to the microcontroller.
5. It can perform basic input-output operations on and button-LED controls microcontroller.
6. It can perform 7 segment display and matrix display applications on microcontroller.
7. It can perform keypad and LCD applications on microcontroller.
8. It can perform analog-to-digital conversion and serial communication applications on microcontrollers.
9. It can learn about the concept of interrupts and their importance in microcontrollers.
10. Students can learn to write more readable and efficient programs by creating their own functions, modularizing their programs, and reducing code repetition.

Weekly Contents

Order	PreparationInfo	Laboratory	TeachingMethods	Theoretical	Practise
1	Students should have basic knowledge of Arduino hardware, fundamental electronic components (LEDs, resistors, etc.), and basic programming structures (setup() and loop() functions). Reference: https://disk.yandex.com/i/V4O8vgQxEj2Scg	Students will install Arduino IDE on their computers, configure the settings, and upload their first "Blink" (LED control) program for testing	The lesson will be supported with visual materials, simulations, and step-by-step exercises covering Arduino IDE, code structures, serial monitor, and basic functions.	This lesson covers the basic structure of Arduino, its differences from microcontrollers, and areas of application. The hardware components of Arduino Uno, input-output pins, and working principles will be examined.	Students will write a simple program using Arduino's digital outputs to turn an LED on and off and observe its operation.

Order	PreparationInfo	Laboratory	TeachingMethods	Theoretical	Practise
2	Students should understand how digitalRead(), digitalWrite(), and delay() functions work and be able to use comparison operators to create decision mechanisms. Reference: https://disk.yandex.com/i/MeIpO7fCV9hMFw	Students will control LEDs using button inputs and test LED blinking applications with the delay() function	The lesson will be supported by visual materials, simulations, and step-by-step exercises, ensuring students learn through hands-on practice.	This lesson will cover the usage of delay() and delayMicrosecond() functions for time delay. Additionally, comparison operators (==, !=, <, >, <=, >=) and their role in decision-making structures in microcontroller applications will be examined.	Students will write programs to turn an LED on when a button is pressed and off when released. Additionally, LED blinking effects will be created using different timing functions.
3	Students should understand how variables are defined, and how if-else decision structures and loops work. Reference: https://disk.yandex.com/i/yfLl7H0Oc1SvyA	Students will implement arithmetic operations and logical decision mechanisms using buttons and LEDs on Arduino.	The lesson will be supported with sample circuits, code blocks, and simulations, ensuring students reinforce their knowledge through practice.	This lesson will cover *arithmetic operators (+, - , , /, %) and logical operators (&&, ||, !) in Arduino. Their functions and applications in programming will be explained.	Students will write programs where pressing a button turns on an LED, and when a specific condition is met, all LEDs turn off.
4	Students should have basic knowledge of variable definitions in Arduino, constant variables (const, #define, volatile), and serial communication principles. Reference: https://disk.yandex.com/i/SPGEuqjow0NqlA	Students will use two buttons in Arduino to control an LED with AND (&&) and OR (||) operators and test how the NOT (!) operator inverts input signals.	The lesson will be supported by sample circuits, simulations, and step-by-step code analysis, ensuring students learn through practice.	This lesson covers the usage and functionality of AND (&&), OR (||), and NOT (!) logical operators. Additionally, the principles of data exchange via Arduino's serial port, baud rate concept, and Serial.print(), Serial.println() functions will be examined.	Students will send a variable from Arduino to a computer using the Serial.print() function and observe the data flow on the serial monitor.
5	Students should have knowledge of variable definitions, comparison operators, and basic input-output operations in Arduino. Reference: https://disk.yandex.com/i/WNk3YBCknqOZLw	Students will control LEDs using different loop structures in Arduino and implement button interactions.	The lesson will be supported with sample codes, circuit designs, and simulations, ensuring that students learn through hands-on experience.	This lesson will cover loop structures in Arduino programming, including the purposes, working principles, and differences between for, while, and do-while loops.	Students will develop applications such as sequential LED blinking with a for loop, waiting until a button is pressed using a while loop, and using a do-while loop for at least one execution.
6	Students should have knowledge of for and while loops, switch-case structures, the digitalWrite() function, and basic input-output operations. Reference-2: https://disk.yandex.com/i/xDSA1Zw2PnZQ9w Reference-2: https://disk.yandex.com/i/EtyEAOM-Cfmo0A	Students will control loops using break and continue statements and test by displaying numbers on a 7-segment display.	The lesson will be supported by sample codes, simulations, and circuit applications, ensuring students experience the differences of break-continue structures and the use of 7-segment displays.	This lesson will cover how break and continue statements are used in loops and switch-case structures. Additionally, the working principles of 7-segment displays, differences between common anode and common cathode, and usage methods will be explained.	Students will develop an application that displays numbers from 0 to 9 on a 7-segment display using Arduino and allows number changes via a button.

Order	PreparationInfo	Laboratory	TeachingMethods	Theoretical	Practise
7	Students should have knowledge of for and while loops, switch-case structures, the digitalWrite() function, and basic input-output operations. Reference-2: https://disk.yandex.com/i/xDSA1Zw2PnZQ9w Reference-2: https://disk.yandex.com/i/EtyEAOM-Cfmo0A	Students will control loops using break and continue statements and test by displaying numbers on a 7-segment display.	The lesson will be supported by sample codes, simulations, and circuit applications, ensuring students experience the differences of break-continue structures and the use of 7-segment displays.	This lesson will cover how break and continue statements are used in loops and switch-case structures. Additionally, the working principles of 7-segment displays, differences between common anode and common cathode, and usage methods will be explained.	Students will develop an application that displays numbers from 0 to 9 on a 7-segment display using Arduino and allows number changes via a button.
8				Ara Sınav	
9	Students should have knowledge of the LiquidCrystal library, lcd.print(), lcd.clear(), lcd.setCursor() commands, and basic serial communication functions. Reference: https://disk.yandex.com/i/XfpKKG7-TBxy8g	Students will connect an LCD display to Arduino and write their first message using lcd.print(), lcd.clear(), and lcd.setCursor() functions.	The lesson will be supported with sample circuit connections, code applications, and simulations, ensuring that students learn LCD usage hands-on.	Students will learn basic information about 2x16 LCD screens.	Students will develop an application that increases or decreases a number displayed on the LCD using a button. Additionally, they will design a football scoreboard system where two teams' scores are controlled via buttons.
10	Students should have basic knowledge of analogRead(), analogReference(), ADC resolution, and analog signal processing. Reference: https://disk.yandex.com/i/aBa4iV7ffNYsyw	Students will measure voltage from a potentiometer or sensor using Arduino's analog inputs and test the ADC conversion with the analogRead() function.	The lesson will be supported by sample circuit connections, voltage calculations, and ADC resolution tests, ensuring that students learn how to read real-world data practically.	This lesson covers the conversion of analog signals to digital, the principle of ADC (Analog-Digital Converter), and the usage of analog inputs in Arduino. The A0-A5 analog input pins, ADC resolution, and the importance of precision will be discussed.	Students will read analog data using a potentiometer or LM35 temperature sensor and display the data on an LCD screen.
11	Students should have basic knowledge of PWM pins in Arduino, the usage of the analogWrite() function, and the concept of duty cycle. Reference-1: https://disk.yandex.com/i/uKUXx4H9HemqHA Reference-2: https://disk.yandex.com/i/ir_IZqB7_8BHUg	Students will use the analogWrite() function to generate a PWM signal and gradually adjust the brightness of an LED. Additionally, the output signal will be analyzed using a multimeter or oscilloscope.	The lesson will be supported by graphical representations of PWM signals, oscilloscope analysis, and interactive simulations, ensuring that students learn how PWM signals are generated and modified through hands-on practice.	This lesson covers the basic principles of Pulse Width Modulation (PWM), duty cycle, frequency, and period relations. The applications of PWM in LED brightness control, motor speed control, and analog signal simulation will be examined.	Students will gradually increase and decrease the brightness of an LED using PWM and create a circuit that controls the speed of a DC motor.
12	Students should have knowledge of the attachInterrupt() function used for implementing interrupts in Arduino, the TimerOne library, and interrupt modes (LOW, CHANGE, RISING, FALLING). Reference: https://disk.yandex.com/i/y_7Zrb70eB1pVQ	Students will use Arduino's INT0 (Pin2) and INT1 (Pin3) interrupt pins to trigger an interrupt upon button press and control an LED accordingly.	The lesson will be supported by simulations, graphical representations, and real hardware applications, allowing students to observe and understand how interrupts function.	This lesson covers the concept of interrupts, their importance, and why they are used in microcontrollers. External Interrupts and Timer Interrupts in Arduino will be explained in detail.	Students will control an LED using an interrupt function triggered by a button and implement a counter application that operates at set intervals using a Timer Interrupt.

Order	PreparationInfo	Laboratory	TeachingMethods	Theoretical	Practise
13	Students should have basic knowledge of function definition in Arduino, parameter passing, return statements, and variable scope (local-global variables). Reference: https://disk.yandex.com/i/50A3ZuvTxeloOw	Students will define a simple function to blink an LED and test how the function can be modified using different parameters.	The lesson will be supported with example function definitions, real-world applications, and simulations, ensuring that students learn how to organize their code efficiently using functions.	This lesson covers how functions are used in Arduino programming, the advantages of modular structure by avoiding code repetition, and different function types. Functions with and without parameters, as well as those that return or do not return values, will be examined.	Students will use functions with and without parameters to control an LED with a button and process data from a temperature sensor using functions.
14	Students should have basic knowledge of function definition in Arduino, parameter passing, return statements, and variable scope (local-global variables). Reference: https://disk.yandex.com/i/50A3ZuvTxeloOw	Students will define a simple function to blink an LED and test how the function can be modified using different parameters.	The lesson will be supported with example function definitions, real-world applications, and simulations, ensuring that students learn how to organize their code efficiently using functions.	This lesson covers how functions are used in Arduino programming, the advantages of modular structure by avoiding code repetition, and different function types. Functions with and without parameters, as well as those that return or do not return values, will be examined.	Students will use functions with and without parameters to control an LED with a button and process data from a temperature sensor using functions.
15	Students should understand array definition rules, how to add and access elements, and how to use arrays within loops. Reference: https://disk.yandex.com/i/wrSThxfjHfXJaw	Students will define one-dimensional and multi-dimensional arrays to test data storage and retrieval. Additionally, they will access array elements and perform assignments within the program.	The lesson will be supported by sample programs, graphical explanations, and simulations, ensuring students learn how to use arrays to write more efficient programs.	This lesson will cover the basic structure of arrays, their importance, and how they differ from variables. Topics such as single-dimensional and multi-dimensional arrays, matrix arrays, and jagged arrays will be discussed.	Students will develop a Knight Rider (walking light) application using LED arrays and implement dynamic LED control through array elements.

Workload

Activities	Number	PLEASE SELECT TWO DISTINCT LANGUAGES
Teorik Ders Anlatım	14	2,00
Uygulama / Pratik	14	2,00
Ev Ödevi	10	2,00
Proje	1	4,00
Ara Sınav Hazırlık	4	1,00
Vize	1	1,00
Final Sınavı Hazırlık	4	1,00
Final	1	1,00

Assesments

Activities	Weight (%)
Final	60,00
Vize	40,00

Elektronik ve Otomasyon Bölümü / KONTROL VE OTOMASYON TEKNOLOJİSİ PROGRAMI X Learning Outcome Relation

	P.O. 1	P.O. 2	P.O. 3	P.O. 4	P.O. 5	P.O. 6	P.O. 7	P.O. 8	P.O. 9	P.O. 10	P.O. 11	P.O. 12	P.O. 13	P.O. 14
L.O. 1	4		5	5	2	4	4	2	1		1	2	5	3
L.O. 2	5		5	5	4	5	3	1	1		1	2	4	2
L.O. 3	4		5	5	4	4	4	2	1		1	2	5	2
L.O. 4	4		5	5	3	5	4	1	1		1	2	5	2
L.O. 5	4		5	5	4	4	3	2	1		1	2	4	2
L.O. 6	4		4	5	3	3	3	2	1		1	2	5	2
L.O. 7	4		4	5	3	2	4	2	1		1	2	5	2
L.O. 8	4		4	5	4	3	4	2	1		1	2	5	2
L.O. 9	4		4	5	4	3	4	2	1		1	2	5	2
L.O. 10	4		4	5	4	4	4	2	1		1	2	5	2

Table :

P.O. 1 : Mesleği ile ilgili temel, güncel ve uygulamalı bilgilere sahip olur.

P.O. 2 : İş sağlığı ve güvenliği, çevre bilinci ve kalite süreçleri hakkında bilgi sahibi olur.

P.O. 3 : Mesleği için güncel gelişmeleri ve uygulamaları takip eder, etkin şekilde kullanır.

P.O. 4 : Mesleği ile ilgili bilişim teknolojilerini (yazılım, program, animasyon vb.) etkin kullanır

P.O. 5 : Mesleki problemleri ve konuları bağımsız olarak analitik ve eleştirel bir yaklaşımla değerlendirme ve çözüm önerisini sunabilme becerisine sahiptir.

P.O. 6 : Bilgi ve beceriler düzeyinde düşüncelerini yazılı ve sözlü iletişim yolu ile etkin biçimde sunabilir, anlaşılır biçimde ifade eder.

P.O. 7 : Alanı ile ilgili uygulamalarda karşılaşılan ve öngörülemeyen karmaşık sorunları çözmek için ekip üyesi olarak sorumluluk alır

P.O. 8 : Kariyer yönetimi ve yaşam boyu öğrenme konularında farkındalığa sahiptir.

P.O. 9 : Alanı ile ilgili verilerin toplanması, uygulanması ve sonuçlarının duyurulması aşamalarında toplumsal, bilimsel, kültürel ve etik değerlere sahiptir.

P.O. 10 : Bir yabancı dili kullanarak alanındaki bilgileri takip eder ve meslektaşları ile iletişim kurar.

P.O. 11 : Kontrol ve otomasyon sistemlerinin temel prensiplerini açıklar, tasarımını ve montajını yapar.

P.O. 12 : Otomasyon sistemlerinde meydana gelebilecek arızaları tespit eder ve arızaları giderir.

P.O. 13 : PLC, mikrodenetleyici ve diğer kontrol sistemleri için yazılım geliştirir.

P.O. 14 : Endüstriyel sensörler ve kontrol elemanları hakkında bilgi sahibi olur ve uygulamalar geliştirir.

L.O. 1 : Mikroişlemciler ile mikrodenetleyiciler arasındaki farkları öğrenebilir.

L.O. 2 : Mikrodenetleyicinin donanım birimlerini ve giriş çıkış birimleriyle işlemler yapabilir.

L.O. 3 : Yüksek ve alçak seviyeli dilleri ve aralarındaki farkları öğrenebilir.

L.O. 4 : C programlama dilini derleyici program vasıtası ile makine diline dönüştürüp mikrodenetleyiciye programı yükleyebilir.

L.O. 5 : Mikrodenetleyicide temel giriş-çıkış işlemleri ve buton-led kontrollerini yapabilir.

L.O. 6 : Mikrodenetleyicide 7 segment display ve matrix display uygulamalarını yapabilir.

L.O. 7 : Mikrodenetleyicide tuş takımı ve LCD uygulamaları yapabilir.

L.O. 8 : Mikrodenetleyicide analog dijital dönüşüm ve seri haberleşme uygulamaları yapabilir.

L.O. 9 : Kesme (Interrupt) kavramını ve mikrodenetleyicilerdeki önemini öğrenebilir.

L.O. 10 : Öğrenciler, kendi fonksiyonlarını oluşturup programlarını modüler hale getirerek ve kod tekrarını azaltarak daha okunabilir ve verimli programlar yazmayı öğrenebilir.