EET-3300: APPLICATIONS PROGRAMMING FOR SMART MANUFACTURING

Cuyahoga Community College

Viewing: EET-3300 : Applications Programming for Smart Manufacturing

Board of Trustees:
September 2023

Academic Term:
Fall 2024

Subject Code
EET - Electrical/Electronic Engineer

Course Number:
3300

Title:
Applications Programming for Smart Manufacturing

Catalog Description:
Programming course with emphasis on smart manufacturing applications. Languages used for embedded and non-embedded environments are covered. These applications require distinguishing between the internal workings of a microprocessor versus communicating over networks. Network communication requires networking protocols. Develop scripts to automate repetitive tasks and improve workflow using the CLI (Command-line interface). Software implementation requires applications performing API calls for both synchronous and asynchronous API (Application Programming Interface) as well as accessing various library routines. This course provides familiarity with universal concepts like data types, containers, functions, conditions, loops semantics, and the runtime environment.

Credit Hour(s):
3

Lecture Hour(s):
1

Lab Hour(s):
6

Requisites

Prerequisite and Corequisite
EET-1600 Industrial Routers, Switches, and Operating Systems for Smart Manufacturing and EET-1620 Industrial Protocols and Machine Connectivity for Smart Manufacturing.

Outcomes

Course Outcome(s):
Explain the internal architecture of a microprocessor with how it relates to communication through to outside an individual computer or individual microcontroller.

Essential Learning Outcome Mapping:
Not Applicable: No Essential Learning Outcomes mapped. This course does not require application-level assignments that demonstrate mastery in any of the Essential Learning Outcomes.

Objective(s):
1. Identify internal special functions registers and memory that includes timers, counters, flash memory, writable memory (Random Access Memory), etc., of a microcontroller and the bus structure of a microprocessor that would access the equivalent of external special functions registers and memory.
2. Explain bus timing and correlate the timing diagram to the minimum and maximum timing specifications.
3. Describe how the internal bus structure of a microcontroller and the external bus structure of a microprocessor transfer data.
4. Explain the three types of buses: address, data and control.

5. Design, develop, debug, execute, and refactor multi-module programs written in different syntax.
6. Analyze and model real-life problems in Object-Oriented programming (OOP) categories.
7. Use the potential programs in everyday applications including IoT and DIY activities.
8. Understand the differences between procedural and OOP approaches and be able to differentiate the pros and cons of both techniques;

Course Outcome(s):

Demonstrate an Integrated Development Environment (IDE).

Essential Learning Outcome Mapping:

Not Applicable: No Essential Learning Outcomes mapped. This course does not require application-level assignments that demonstrate mastery in any of the Essential Learning Outcomes.

Objective(s):

1. In the IDE, create a new project and specify the family and part that the project will use.
2. Create source programs in a programming language and add it to the project.
3. Include library routines in the programs.
4. Compile the programs.
5. Correct syntax errors that the compiler identifies.
6. Demonstrate and use the IDE's debugger functions that include breakpoints, single step, variable contents, etc.
7. If present, find and correct run time errors.

Course Outcome(s):

Demonstrate the three principal components of a programming language.

Essential Learning Outcome Mapping:

Not Applicable: No Essential Learning Outcomes mapped. This course does not require application-level assignments that demonstrate mastery in any of the Essential Learning Outcomes.

Objective(s):

1. Write and test programs using selector statements that include if and if-else statements.
2. Write and test programs that use iteration statements such as for loop statements.
3. Write and test programs that use process statements and explain how their execution is controlled by selector and loop statements.

Course Outcome(s):

Implement API requests and calls (Application Programming Interface).

Essential Learning Outcome Mapping:

Not Applicable: No Essential Learning Outcomes mapped. This course does not require application-level assignments that demonstrate mastery in any of the Essential Learning Outcomes.

Objective(s):

1. Implement both synchronous and asynchronous API (Application Programming Interface) requests.
2. Use software development tools to test the API calls.

Course Outcome(s):

Execute methods to automate repetitive tasks.

Essential Learning Outcome Mapping:

Not Applicable: No Essential Learning Outcomes mapped. This course does not require application-level assignments that demonstrate mastery in any of the Essential Learning Outcomes.

Objective(s):

1. Define workflows for doing configuration, verification, and monitoring.
2. Demonstrate use of the CLI (Command-line interface) to implement scripts.
3. Implement running software tools that require including library routines.

Course Outcome(s):

Using software applications, read data from ports and write data to ports.

Essential Learning Outcome Mapping:

Not Applicable: No Essential Learning Outcomes mapped. This course does not require application-level assignments that demonstrate mastery in any of the Essential Learning Outcomes.

Objective(s):

1. Configure ports for input and demonstrate how data is transferred from a port to a variable and prove the variable has the correct data.
2. Configure ports for output and transfer data from a variable to an output port and prove the correct data is on the port.
3. Describe how the same port can in some applications both input and output data.

Methods of Evaluation:

a. Examinations

b. Quizzes

c. Lab reports

d. Homework assignments

Course Content Outline:

1. Microprocessors and internal communication of an individual computer vs. network communication with its protocols
   a. Internal data, control, and address buses
   b. Network communication through ports and communication channels
2. Code address space vs. Data address space
   a. Encoding (including if Big-Endian or Little-Endian) of addresses
   b. Distinguishing code from data
3. Addresses mapped for I/O
   a. Set specific addresses as input or output
   b. Read from address or write to address, depending on how I/O is set
4. Arithmetic and logic units (ALU)
   a. Performing addition and subtraction
   b. Performing multiplication and division
5. Ports
   a. GPIO (General Purpose Input/Output)
   b. USB (Universal Serial Bus)
6. Flags
   a. Logic flag
   b. Mathematical operation flags
7. Independent Development Environment (IDE)
   a. Editor in IDE
   b. Display of syntaz errors in IDE
8. Including library routines
   a. Math library
   b. Input-output libraries
9. API requests and calls (Application Programming Interface)
   a. Synchronous API calls
   b. Asynchronous API calls
10. Software development tools to test the API calls
    a. API client
    b. API endpoint
11. Automation of repetitive tasks

   a. Automation using scripts
   b. Other types of automation
12. CLI (Command-line interface)
   a. Windows CLI
   b. Linux CLI
13. Scripts
   a. Scripts commands
   b. Script keywords and arguments
14. Workflows for doing configuration, verification, and monitoring
   a. Workflow actions
   b. Workflow conditions

Resources

Louis E. Frenzel. *Principles of Electronic Communication Systems*. Fourth. New York, NY: McGraw-Hill Education, 2016.

Eric Chou. *Mastering Python Networking: Utilize Python packages and frameworks for network automation, monitoring, cloud, and management*. Fourth. Birmingham, England: Packt Publishing, 2023. https://www.amazon.com/Mastering-Python-Networking-frameworks-automation/dp/180323461X/ref=sr_1_4_sspa?crid=32FHVUCWG1J41 Mastering Python Networking: Utilize Python packages and frameworks for network automation, monitoring, cloud, and management, 4th Edition: 9781803234618: Computer Science Books @ Amazon.com

---

Hans-Petter Halvorsen. *Python for Control Engineering*. First. The Technical Guy-a Blog about Technology, 2020. April 25, 2022. https://www.halvorsen.blog

---

Tony Gaddis. *Starting Out with C++*. Ninth. New York, New York: Pearson, 2019.

Brain Kernighan. *Programming in C*. Second. Hoboken, New Jersey: Prentice Hall, 1988.

Resources Other

a. Lab packet

b. Handouts

   i. Instructor authored notes
   ii. Topic summaries
   iii. Journal articles
   iv. Manufacturer's literature

Top of page

Key: 5140