MECHANICAL ENGINEERING PROGRAM
ABET COURSE SYLLABUS

ME 305 Mechatronics (4 Units) Required for Mechatronics Concentration, Elective for all others

Course Description: (2019-20 Catalog)	Introduction to microcontrollers and assembly language programming. Emphasis on components and techniques for interfacing that are typical of embedded microcontroler applications (A/D conversion, D/A conversion, interrupts, timers, and pulse-width modulation). Laboratory exercises involve real-time interfacing of microcontrollers to external mechanical and/or electromechanical devices. 3 lectures, 1 laboratory.
Prerequisite Courses:	EE 201 and EE 251, or consent of instructor.
Prerequisites by Topic:	Analytical skills and lab familiarity with basic analog electronics.
Optional Textbook:	**The HCS12/9S12: An Introduction to Software and Hardware Interfacing, 2nd ed.**, by H. Huang, Cengage, 2010.
References:	**Microcontroller Theory and Applications: HC12 & S12**, **2nd ed.**, by D.J. Pack and S.F. Barrett, Prentice Hall, 2008. **Introduction to Mechatronic Design,** by J.E. Carryer, R.M. Ohline, and T.W. Kenny, Prentice Hall, 2011. Microcontroller manufacturer's hardware and software reference manuals. Data sheets from electronic component manufacturers.
Course Coordinator/Instructor:	William R. Murray, Professor, Mechanical Engineering
Course Learning Outcomes:	1. The student can analyze real-time constraints and design software to meet them. 2. The student can use reference materials to find the detailed information necessary to interface a microcontroller to an external device. 3. The student can design a moderately complex real-time program using an organized design methodology. 4. The student is skilled at writing and debugging program code. 5. The student documents program designs thoroughly and accurately. 6. The student works effectively as a member of a development team.

Relationship of Course to Mechanical Engineering Student Outcomes:

SO 1: Develop (D)
SO 2:
SO 3: Develop (D)
SO 4:
SO 5: Develop (D)
SO 6:
SO 7:

Topics Covered:

1. Number systems and binary arithmetic
2. Microcontroller architecture and instruction set
3. The task-state-transition software design method
4. Programming in assembly language
5. Microcontroller interfacing: digital I/O, A/D conversion and D/A conversion
6. Hardware and software timing and interrupts
7. Techniques for effective testing and debugging of real-time embedded code

Laboratory Projects:

1. Lab 1: Software Development System — An introduction to the microcontroller and software development system. 1 week.
2. Lab 2: Finite State Machine Programming — Interface microcontroller to drive two sets of LEDs at independent rates with code structured as independent cooperative tasks, with each task structured as a finite state machine. 1 week.
3. Lab 3: I/O and Binary-Coded-Decimal Conversion — Follow-on to the previous lab with the addition of keypad input, LCD output, and real-time user interface. 3 weeks.
4. Lab 4: Function Generator — Combine skills learned in previous labs to design and implement a function generator program with multiple waveforms, keypad input, LCD output, and user interface. 2 weeks.
5. Lab 5: Motor Controller — Design and implement a PID controller to control the velocity of a dc servomotor equipped with an optical encoder, including keypad input, LCD output, and real-time user interface. Comparison between simulated response and experimental results required. 3 weeks.

Class/Lab Schedule: Three 50-minute lectures per week, one 170-minute lab per week.

Contribution of Course to Meeting the Professional Component:

(a) College-level mathematics and basic sciences:	0 credits
(b) Engineering Topics:	4 credits
Design?	Yes
(c) General Education:	0 credits
(d) Other:	0 credits

Prepared by: William R. Murray **Date:** 07/02/2020.