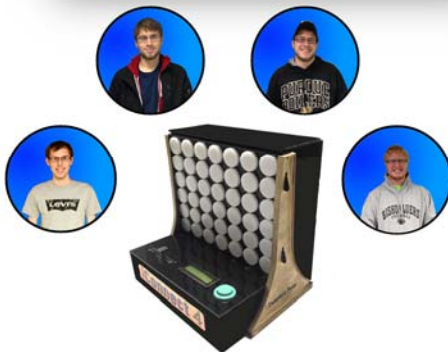
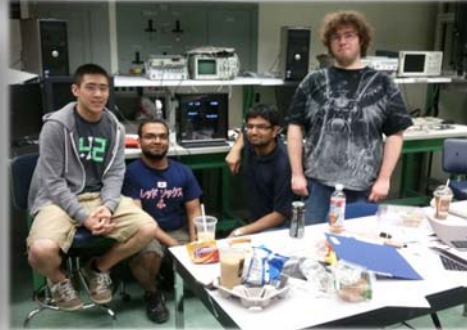
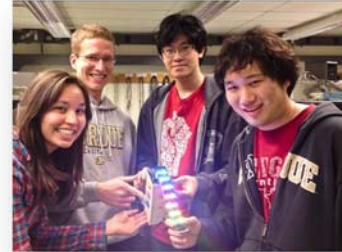
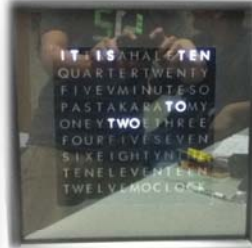
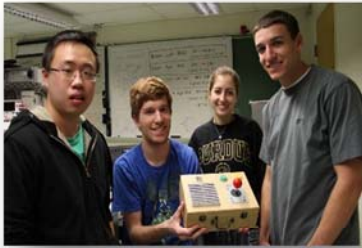


ECE 362 – Microprocessor Systems and Interfacing

Course Syllabus



PURDUE
UNIVERSITY
School of Electrical & Computer Engineering

COURSE POLICIES AND PROCEDURES

Course Description: An introduction to control-oriented microprocessor (“microcontroller”) architecture, software, interfacing, and peripherals, with an emphasis on basic computer engineering concepts. This is *not* a course about “personal” or general-purpose computers, but rather about embedded microcontrollers that serve as the *basic building block* of a wide variety of “intelligent” products.

Purpose of Course: ECE 362 provides an introduction to microprocessors, assembly language programming techniques, interface hardware design, common on-chip peripheral devices used in microcontroller-based systems, and general computer engineering concepts. As such, this course relies *heavily* on background obtained in the required prerequisite courses (listed below). Since it is geared toward meeting core ECE curriculum requirements, this course is *not recommended* for students having little or no engineering or electronics background.

Required Background: Required prerequisites are ECE 270 and CS 159 (or equivalent). Further, it is *highly recommended* that these courses be taken *recently*.

Course Web Site URL: <https://engineering.purdue.edu/ece362>

Course E-mail Address: ece362@ecn.purdue.edu

Materials Required for Purchase:

1. Motorola 9S12C32 Microcontroller Kit, available on-line from Technological Arts
2. DK-3 Parts Kit (Part No. 32PRDWLFDK3), available on-line from Electronix Express
3. “iClicker” Student Response Unit (available at local bookstores)

Lecture Notes: Posted on the course web site. Each student will be responsible for printing his/her own copy of these materials and bringing them to class.

Computer Account: If you don’t already have one, you *must* obtain a “coordinated” ITaP/ECN computer account to use the machines in lab. Requests can be sent to ecsite@ecn.purdue.edu. Further, you *must* establish a “*name@purdue.edu*” e-mail alias so that course information can be automatically sent to you.

Consultation Outside of Class: Scheduled office hours for all course staff members will be posted on the course web site. You are encouraged to make good use of all the “live” office hours available for this course – please do not E-mail long/detailed questions to the course staff about the lecture/lab material, as this type of communication tends to be very inefficient.

Class Number: The identifier you will use on all materials turned in for this course, referred to as your *class number*, will be constructed by concatenating the *last four digits* of your *student identification number* with the *first character* of your *last name*. For example, if your PUID number is “01234-56789” and your last name is “Jones”, your *class number* would be “6789-J”.

Class Attendance: In a word, **REQUIRED**. You must be **physically present** to earn the associated class participation credit – no substitution will be allowed! ***It will be considered “cheating” for someone to use the iClicker of another student to “fake” that student’s class participation.***

Homework: Assigned homework exercises will be checked for completion in lab (see course website for assignments and due dates).

Laboratory: The weekly lab meetings for this course will be held in room EE 056. The lab experiments have been designed to reinforce the lecture material; thus, it is very important that you ***attend lecture regularly and do the assigned reading from the provided references*** in order to successfully complete the laboratory portion of this course. A separate document details the *Laboratory Policies and Procedures*.

Changing Lab Divisions: You must consistently attend the lab division of ECE 362L for which you have registered. All lab division changes must be finalized during the first week of classes.

Learning Outcomes and Objectives: A student who successfully fulfills the course requirements will have demonstrated:

1. an ability to program a microcontroller to perform various tasks
2. an ability to interface a microcontroller to various devices
3. an ability to effectively utilize microcontroller peripherals
4. an ability to design and implement a microcontroller-based embedded system

Learning Outcome Assessment: ***You will earn 1% bonus credit for each course outcome you successfully demonstrate.*** Outcome 1 will be assessed based on scores received for the lab practical programming problems, for which a ***score of at least 60%*** on either of the two practical exams will be required or a ***score of at least 60%*** on each lab experiment to demonstrate basic competency. For Outcomes 2 and 3, basic competency will be assessed based on lab practical concept exams during your scheduled lab period, for which a ***score of at least 60%*** will be required to demonstrate basic competency. Outcome 4 will be assessed based on the Embedded System Design Mini-Project completed, for which a ***score of at least 60%*** will be required to demonstrate basic competency.

Exam Calculator Policy: Exams in this course will be designed such that stand-alone calculators are not required and, therefore, will be prohibited (you will, however, be allowed to use the Windows calculator application).

Makeup Exams: If you have an excused absence from a scheduled exam based on documented participation in a University-related activity (e.g., band trip to “enemy territory”), you must make arrangements **in advance** to take your exam during an alternate time (lab period) during the week the exam is being administered. If absence during a scheduled exam period is excused (due to illness or family emergency, which must be verified ***in writing***), you will be given the opportunity make up the missed exam during finals week.

Embedded System Design Mini-Project: Based on the premise that the “best way to learn about a topic is to write your *own* question and then proceed to answer it,” a significant portion of the course will be devoted to completing an embedded system design of your own choosing. The basic requirement is to design a product based on the Motorola 9S12C Microcontroller Demo Kit

that makes effective use of the processor's computational and interfacing resources. Project teams consisting of 3-4 students each will be formed by mid-semester, and project proposals will be due shortly thereafter. Project demonstrations will be scheduled on a team-by-team basis during the last two weeks of lab. Details about the Mini-Project requirements and grading criteria can be found on the course web site.

Grade Weightings: Your raw score (i.e., "**Raw Weighted Percentage**" or **RWP**) will be calculated based on the following weights:

Bonus Exercises	$\Delta_1\%$
Class Participation (iClickers)	5.0%
Homework Completion and Lab Notebook Maintenance	10.0%
Lab Experiments (10 @ 2%)	20.0%
Lab Quizzes (10 @ 0.5%)	5.0%
Lab Practical Concept Exams – Outcomes 2 & 3 (2 @ 15%)	30.0%
Lab Practical Programming Exams – Outcome 1 (2 @ 7.5%)	15.0%
Embedded System Design Mini-Project – Outcome 4	15.0%
Outcome Demonstration Bonus (4 Outcomes @ 1%)	$\Delta_2\%$
	<hr/>
	100+ $\Delta\%$

Course Grade Determination: Your **Raw Weighted Percentage (RWP)**, described above, will be mean-shifted with respect to the upper percentile of the class to obtain a **Normalized Weighted Percentage (NWP)**. For example, if the top student has an RWP of 97%, everyone's RWP will be multiplied by 1.031 to obtain their corresponding NWP. Equal-width cutoffs will then be applied based on the **Windowed Standard Deviation (WSD)** of the raw class scores; the minimum **Cutoff Width Factor (CWF)** used will be 10 (i.e., the nominal cutoffs for A-B-C-D will be 90-80-70-60, respectively). Letter grades in the upper 30% of each range will have a "+" designation, and those that fall in the lower 30% of each range will have a "-" designation.

Reporting of Projected Course Grades: Your projected course grade will be calculated periodically throughout the semester and reported to you via E-mail. Final course grades will be reported to you in a similar fashion.

Borderline Cases: A "borderline" is officially defined as an NWP within 0.5% of a cutoff when the final grade calculation is performed. Before course grades are assigned, the instructor will carefully examine all such cases and determine if the next higher grade is warranted. A special bonus, IDPPB (*Instructor Discretion Posi-Points Bonus*), will be used to facilitate borderline adjustments; **note, however, that the "next higher grade" is NOT automatically guaranteed.**

Incompletes and Conditional Failures: A grade of "I" or "E" will be given *only* for cases in which there are **documented** medical or family emergencies that prevent a student from completing

required course work by the end of the semester. Note that University Regulations stipulate that a student must be passing in order to **qualify** for a grade of “I” or “E”.

Campus Emergencies: In the event of a major campus emergency, course requirements, deadlines, and grading percentages are subject to changes that may be necessitated by a revised semester calendar or other circumstances beyond the instructor’s control. Should such an emergency occur, information will be posted on the course web site Message Board.

“The Best Way to Study for This Course”: The best way to learn the kind of material covered in this course is to **review it as soon after attending class as possible** and to **practice similar problems**. The Lecture Workbook will help you collect well-organized notes as well as help you **actively encode** essential course material (instructional research has shown that notes **you** take in class serve more than for mere “archival” of information – they actually help you **remember** the material presented in lecture). Obviously, *regular lecture attendance* is essential to the entire process. Further, given that we typically *forget* 90% of what we *hear*, but *remember* 80% of what we do, the “best way to study for this course” can be summed up in three words: practice, practice, practice! **The keys to success are keeping current with the material and making effective use of the learning resources available (live office hours, lecture videos, posted references and solutions, etc.).**

Professionalism and Academic Honesty: The temptation to cheat is particularly prevalent in large enrollment courses such as this one. In the long run, *short-cuts in school work* breed *short-cuts in careers*, i.e., the less you invest in your education, the less you will have to show for it later in life. A large part of the educational process is simply developing the *discipline* and *mindset* required to contribute in a given technical area once you graduate. If for nothing other than your own benefit, *do not copy the work of any other student* (past or present). Further, be advised that any *documented* case of “cheating” will result in a **FAILING GRADE** for the course as well as possible disciplinary action. All cases of academic dishonesty will be reported to the ECE Associate Head as well as to the Dean of Students. **A professional person does not take credit for the work of someone else.**

Examples of Cheating: Contrary to the beliefs of the post-modern “situational ethics” crowd, there are indeed absolutes that apply to integrity and honesty. Examples of activities that will be construed to be “cheating” include **(but are not limited to)** the following:

- copying the work of another student (past or present) and representing that work as your own
- divulging the contents of an exam to students who have not yet taken it
- obtaining information about an exam prior to taking it
- having someone else take an exam for you
- bringing “cheat sheets” in any shape/form with you to an exam
- using a cell phone or other electronic device to share information during an exam
- using a pen camera, cell phone, or any other device to photograph exam materials
- modifying a graded lab or homework paper and submitting it for reevaluation
- sharing lab solutions with other students
- using another student’s (past or present) lab or homework files
- discussing the lab practical exam before all students have completed it

- using an iClicker that belongs to someone else to “fake” their class attendance
- copying and/or redistributing any of the copyrighted materials posted on the course web site
- posting solutions to homework problems or lab experiments

Mediocrity: One of the biggest problems facing our educational system today is mediocrity – too many students want to learn merely (often *barely*) enough to “get by”. In fact, higher education is probably the only “commodity” from which consumers want to get the *least* amount for their money! Clearly, if our country is to be competitive in an expanding world market, a renewed commitment to excellence is absolutely essential. To quote C. R. Swindoll, “The greatest waste of natural resources is the number of people who never achieve their potential.”

Bonus Exercises: Good for you if you’ve read this far! There will be numerous exercises that will count as bonus credit toward your course grade. To earn bonus credit, you must: (1) consistently attend class, (2) bring notes and reference materials with you to each class meeting, (3) complete reading any assigned text/reference material before each class meeting, and (4) regularly check the *Message Board* on the course web site.

Lecture Outline:

<i>Week(s)</i>	<i>Lecture Topics</i>
1-4	Software: microcontroller instruction set and assembly language programming techniques
5-8	Interfacing: microcontroller bus timing analysis, multiplexed bus expansion, general-purpose I/O and printer interface, buffered I/O handling, interrupt handling, applications
9-12	Peripherals: real-time interrupt (RTI), analog-to-digital converter (ATD), serial peripheral interface (SPI), timer module (TIM) and pulse accumulator (PA), pulse width modulation (PWM), serial communications interface (SCI)
13-14	Embedded system design considerations: external peripherals, power supply design, printed circuit board (PCB) design, hardware/software debugging tips
15	Course summary and evaluation, Mini-Project Showcase

Lab Outline:

<i>Week(s)</i>	<i>Lab Experiments</i>
1	Lab Organization
2	Introduction to Development Environment
3	Microcontroller Instruction Set Architecture
4	Assembly Language Programming Techniques
5	Modular Programming Application
6	Bus Timing Visualization Using a Logic Analyzer
7	Interrupt and Device Flag Application
8	Midterm Lab Practical (Outcomes 1&2)
9	RTI and ATD Application
10	TIM and SCI Application
11	PWM and PA Application
12	ATD, TIM, and PWM Audio Application
13	Final Lab Practical (Outcomes 1&3)
14-15	Embedded System Design Mini-Project (Outcome 4)

LABORATORY POLICIES AND PROCEDURES

Lab Policies:

- Students must consistently attend the lab division for which they have registered
- Lab division changes must be finalized during the first week of classes
- Use of lab facilities is restricted to students currently enrolled in the course – **students should not share the lab access code with anyone not enrolled in the course**
- All documents pertaining to lab will be posted on the course web site
- It is the responsibility of each student to print his/her own copy of the lab documents provided on-line – this should be done **in advance** of their scheduled lab meeting
- Abuse of lab printers and waste of paper will **not be tolerated** — failure to abide by these rules will result in loss of printing privileges:
 - ❑ only print **course-related** materials on lab printers
 - ❑ promptly **pick up** all printed output
 - ❑ promptly **recycle** discarded output
 - ❑ ask your lab instructor for assistance if printer jams
 - ❑ load printer with an **entire ream** of paper when printer runs out (do not leave partial reams of paper sitting out on table)

Rules and Regulations:

- **No food, drink, or smoking** is permitted in the lab
- **Do NOT use staples** – students should place course materials directly into their *Lab Manual* in a “loose leaf” fashion
- **Do NOT work alone – only use the lab when another student is present. Failure to follow this rule will result in loss of access to the lab outside of scheduled lab hours.**
- Lab stations must be **returned to their original condition** before students leave
- Any programmable logic devices used for an experiment must be **erased** after the solution has been demonstrated to the lab instructor
- Students are NOT permitted to “snoop” around in cabinets for parts, tools, or equipment
- No parts, tools, or equipment may be removed from lab
- Writing on or otherwise defacing the lab equipment or furnishings will result in disciplinary action
- Wet umbrellas should NOT be brought into lab – please leave them in the hallway to dry

Lab Experiment Grading:

- Identified “**pre-lab**” portions of an experiment must be available for evaluation at the **beginning** of the scheduled lab period
- Steps of experiments must be demonstrated to the lab instructor **as they are completed**
- All work for an experiment must be completed and verified **by the end of the student’s scheduled lab period**
- **Each** student must complete and verify their **own** copy of the assigned work – **electronic or photo copies of lab experiments, code listings, or homework exercises are not acceptable**
- **No credit will be awarded for turning in a copy of another student’s work** (this will be considered “**cheating**”)

Lab Office Hours:

- Beginning the second week of classes, scheduled times are Monday, Tuesday, Wednesday, and Thursday evenings (**except on official University holidays**), 7:00 PM – 10:00 PM
- Students are encouraged to make use of lab office hours to seek help with completing pre-labs; however, they are **still expected to attend their regularly scheduled lab period** to take the quiz, verify completion of the experiment steps, and submit the completed work
- Use of lab office hours to make up a missed experiment **must be approved in advance** by the student's lab instructor

Lab Make-ups:

- Attendance during a student's scheduled lab period is **required** unless he/she has submitted **documentation** verifying an **officially excused absence** (note – "I slept through my alarm clock" or "I had an exam in another class" does NOT constitute an "officially excused absence")
- All experiments must be completed during the scheduled lab period
- All requests for makeup labs must be approved **in advance** (of the evening office hour session the student plans to complete the makeup) – **a documented reason for the absence will be required**
- **No credit will be awarded for makeup labs that have not been officially pre-approved**
- Make-ups **must be completed within one week of the missed lab**
- A make-up lab **must** be completed during **one** lab office hour session (i.e., it cannot be completed over **multiple** sessions)

Quizzes:

- Quizzes will be given at the beginning of each scheduled lab period (except during lab practical exams)
- **There will be no makeup quizzes – if a student fails to show up at the beginning of his/her scheduled lab period, they will not be permitted to take the quiz for that week**
- Provided a requested makeup lab has been **approved** (based on submission of an **officially excused absence**), the corresponding quiz that was missed will be pro-rated (**limit of 2**)

Lab Notebooks:

- Students should obtain a large (two-inch) 3-ring binder in which to create their *Lab Notebook* containing lecture summary notes, completed lab experiments, completed homework, and pertinent reference documents
- **Divider pages** (with **labeled "tabs"**) should be used to organize the material in a meaningful fashion (**note – "post it notes" are not acceptable tabs**)
- Documents within the *Lab Notebook* should be "loose leaf" (i.e. **they should not be stapled together**)
- A carefully maintained *Lab Notebook* will prove invaluable when preparing for exams, lab experiments, and future courses which build on the material covered in ECE 362

LEARNING OUTCOMES AND OBJECTIVES

A student who successfully fulfills the course requirements will have demonstrated:

1. an ability to program a microcontroller to perform various tasks [e,k]

- 1-1. list differences in “world views” regarding the role of microprocessors
- 1-2. define characteristics that distinguish microprocessors
- 1-3. describe the Freescale 68HC(S)12 architecture and programming model
- 1-4. identify different types of memory and describe how each is used
- 1-5. identify instruction addressing modes and syntax
- 1-6. describe the key characteristics of a microprocessor programming model
- 1-7. list microprocessor instruction groups and classify machine instructions accordingly
- 1-8. determine instruction encoding formats and execution cycle counts
- 1-9. determine the effective address of an operand based on the addressing mode used
- 1-10. describe the operation of the stack and identify the instructions that manipulate it
- 1-11. analyze (trace) the execution of assembly code programs
- 1-12. determine how the condition code register is affected by various arithmetic group instructions
- 1-13. list commonly used pseudo-ops and describe their purpose
- 1-14. describe how expressions can be utilized in assembly language code
- 1-15. list the types of constants that can be utilized and describe how they are specified
- 1-16. describe the utility of labels
- 1-17. describe how if-then-else and case control structures can be realized in assembly code
- 1-18. distinguish among and effectively utilize for, do, repeat, and while loop control structures
- 1-19. compile C language code segments into assembly language
- 1-20. compare “hand compiled” C code segments with machine-compiled code
- 1-21. calculate cycle counts of various loop structures and compare their effectiveness
- 1-22. identify potential applications of table lookup and effectively utilize them
- 1-23. distinguish among and effectively utilize different parameter passing techniques
- 1-24. investigate the effect of an indefinite arithmetic result (overflow, divide-by-zero)
- 1-25. define a macro and describe its utility
- 1-26. compare and contrast macros with subroutines
- 1-27. describe the function and utility of conditional assembly
- 1-28. determine the fields of an object file (S-Record)
- 1-29. create an assembly language or C program that performs a prescribed task
- 1-30. diagnose and correct (debug) programming errors

2. an ability to interface a microcontroller to various devices [a,b,c,e,k]

- 2-1. identify CPU and memory timing parameters
- 2-2. draw a bus timing diagram for a simplex CPU-memory interface
- 2-3. identify the critical read and write cycle paths on a bus timing diagram
- 2-4. draw a timing diagram for a multiplexed bus CPU-memory interface

- 2-5. define setup and hold times as they pertain to a CPU-memory interface and discuss the consequences of violating them
 - 2-6. calculate read and write timing margins for a given CPU-memory combination
 - 2-7. describe the potential consequences of excessive read float delay
 - 2-8. define timing margin and discuss the consequences of an insufficient margin
 - 2-9. design a memory expansion interface for a multiplexed bus microcontroller
 - 2-10. determine the nature and extent of bus fighting that occurs in a CPU-memory interface
 - 2-11. evaluate the viability of a CPU-memory interface on the basis of read and write timing margins provided as well as the potential for bus fighting
 - 2-12. diagnose and correct insufficient timing margin in a CPU-memory interface
 - 2-13. diagnose and correct excessive bus fighting that occurs in a CPU-memory interface
 - 2-14. compare the performance of a multiplexed bus memory expansion interface with and without stretch cycles
 - 2-15. evaluate the appropriateness of a memory expansion interface (with and without stretch cycles) based on the address reference mix of a particular application
 - 2-16. identify the key characteristics of an I/O port pin
 - 2-17. describe the operation of a port pin “pull” (up/down) device
 - 2-18. distinguish between “full” and “reduced” drive mode of a port pin with respect to voltage swing and current sourcing/sinking capability
 - 2-19. cite an example of I/O device communication handshaking protocol and create a device driver routine that implements it
 - 2-20. describe the operation of a FIFO (circular) buffer and outline the steps of producer and consumer processes that manipulate it
 - 2-21. define an interrupt and describe how a processor responds when one occurs
 - 2-22. distinguish between maskable and non-maskable interrupts
 - 2-23. define the deterministic and non-deterministic components of interrupt servicing latency
 - 2-24. create an interrupt service routine that performs a prescribed task
- 3. an ability to effectively utilize microcontroller peripherals [j,k]**
- 3-1. describe the key components of the CRG (clock and reset generation) block
 - 3-2. determine the PLL (phase locked loop) settings needed to operate the processor at a prescribed clock frequency
 - 3-3. describe the operation and utility of the RTI (real time interrupt) module of a microcontroller
 - 3-4. configure the RTI subsystem to generate a prescribed interrupt rate
 - 3-5. analyze the accuracy of timing applications based on the RTI subsystem
 - 3-6. create an interrupt service routine that handles an RTI interrupt for a prescribed application
 - 3-7. discuss key design considerations of a microcontroller-based data acquisition system: sampling, quantizing, and encoding
 - 3-8. determine the sampling rate required (Nyquist rate) based on input signal spectral characteristics
 - 3-9. describe aliasing and discuss how it can be prevented

- 3-10. estimate the dynamic range required based on converter resolution and calculate signal-to-quantizing-noise-ratio (SQNR)
- 3-11. describe the operation of a successive approximation analog-to-digital (ATD) converter
- 3-12. identify ATD features and operating modes
- 3-13. configure the ATD module to operate in a prescribed mode
- 3-14. create an ATD device driver routine for a prescribed application
- 3-15. describe the key features of synchronous serial communication
- 3-16. identify SPI features and operating modes
- 3-17. distinguish among MISO, MOSI, SISO, and MOMI entities as they pertain to the SPI module
- 3-18. configure the SPI to operate in a prescribed mode
- 3-19. create an SPI device driver routine for a prescribed application
- 3-20. describe the function of the TIM input capture mechanism
- 3-21. describe the function of the TIM output compare mechanism
- 3-22. describe the function of the TIM pulse accumulator mechanism
- 3-23. distinguish among input capture, output compare, and pulse accumulator timer applications
- 3-24. identify TIM features and operating modes
- 3-25. configure the TIM to operate in a prescribed mode
- 3-26. create a TIM device driver routine for a prescribed application
- 3-27. discuss why the TIM module can serve as a higher precision reference for timer applications than the RTI subsystem
- 3-28. define pulse width modulation (PWM) and describe potential applications
- 3-29. define natural sampling and contrast it with pulse-code modulation (PCM)
- 3-30. measure and compare the effects of different input and output sampling frequencies on signal integrity in an ATD-PWM digitization loop
- 3-31. describe how a PWM channel can be used as a digital-to-analog (DTA) converter
- 3-32. identify PWM features and operating modes
- 3-33. configure the PWM to operate in a prescribed mode
- 3-34. describe asynchronous serial communication character frame format and justify the need for start/stop bits
- 3-35. compare non-return-to-zero (NRZ) serial data encoding with Manchester encoding
- 3-36. describe the difference between half-duplex with full-duplex communication
- 3-37. define baud rate and describe its relation to the local clock frequency
- 3-38. diagnose potential errors that can occur in asynchronous serial communications (parity, framing, overrun) and describe how they can be corrected
- 3-39. distinguish between single-ended and differential serial interfaces, and cite examples of each
- 3-40. describe the purpose and operation of a level translator (line driver) chip
- 3-41. identify SCI features and operating modes
- 3-42. configure the SCI to operate in a prescribed mode
- 3-43. create an SCI device driver routine for a prescribed application

4. an ability to design and implement a microcontroller-based embedded system [a,d,e,f,h,k]

- 4-1. design an interface that allows a microcontroller port pin to control a D.C. load (e.g., motor)
- 4-2. describe how optical isolation can be used to protect microcontroller port pins used as either inputs or outputs
- 4-3. describe how a scanned keypad works
- 4-4. describe the operation of a rotary pulse generator (RPG) and cite some potential applications
- 4-5. describe how a stepper motor works and be able to distinguish between full- and half-step modes
- 4-6. describe how port expansion can be accomplished using shift registers
- 4-7. design an interface that allows a microcontroller to communication with an LCD
- 4-8. design a regulated power supply
- 4-9. form a project team consisting of 2-4 students
- 4-10. choose a project idea and write a proposal
- 4-11. design the hardware and software necessary to realize the project idea
- 4-12. design, fabricate, and test a printed circuit board (honors contract)
- 4-13. package the finished system
- 4-14. demonstrate the project's functionality to the course staff
- 4-15. write a formal report documenting the embedded system design process