Microprocessor Systems and Interfacing

ECE 362 Course Introduction

https://engineering.purdue.edu/ece362/
Three Important Things To Do Today

- Purchase a lab development kit
- Sign up for Piazza
- Purchase a textbook

Details are on the ECE 362 website

https://engineering.purdue.edu/ece362
Your Instructor’s Contact Information

E-mail: rick@purdue.edu
Office: EE 252
Hours: TBD
https://engineering.purdue.edu/ece362/
How did I get that e-mail address?

- Look at [https://directory.purdue.edu/](https://directory.purdue.edu/)
- Click on icon to UPDATE MY DIRECTORY LISTING.
How did I get that e-mail address?

- You still keep the e-mail address that was picked for you.
  - You can still send e-mail to "kennell" if you spell it right.

- Add an e-mail address that’s something you actually want.
  - Can be more than 8 characters.
  - Choose responsibly.
Course Description

- Introduction to (“small memory model”) control-oriented microcontroller software, hardware, and interfacing
- Emphasis: basic computer engineering concepts
- Not a course about “personal” or general-purpose computers, but rather about embedded microcontrollers
Purpose

• To provide an introduction to microcontrollers, assembly language programming techniques, interface hardware design, embedded system design, and general computer engineering concepts
• Specifically geared toward meeting core ECE curriculum requirements
Where ECE 362 fits within the broader computer engineering curriculum
Prerequisites

- Course on high-level language programming
  - (Purdue equivalent: CS 159)
- Course on digital logic design
  - (Purdue equivalent: ECE 270)
Why This Course Is Important

- Embedded microcontrollers are used extensively in process control, instrumentation, home appliances, automobiles, etc. – they represent a **basic building block** of modern digital systems design and the future “Internet of Things” (IoT)
- If you go into virtually any form of engineering design, there is a high probability that knowledge of embedded microcontrollers will be required
Course Text, E-mail, and Web Site

REFERENCE TEXT:


- Lecture notes will also be on the course web site.
- You will also need to purchase an iClicker student response unit, available at bookstores (register your iClicker on Blackboard)
- Please E-mail all course-related correspondence to ece362@ecn.purdue.edu
- Course web site – “everything you need to succeed in ECE 362 is posted here” – [https://engineering.purdue.edu/ece362/](https://engineering.purdue.edu/ece362/)
Lab Kits and References

You will need to purchase an **STM32F01 Discovery Kit** (which includes a variety of interfacing components) from **Technological Arts** during your **first lab meeting (this week)**

- Follow the link provided on course website Message Board – **FREE SHIPPING if ordered this week!**

**PROVIDED ON COURSE WEBSITE:**

- Numerous ARM reference documents
<table>
<thead>
<tr>
<th>Monday</th>
<th>Tuesday</th>
<th>Wednesday</th>
<th>Thursday</th>
<th>Friday</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Jan 7</strong></td>
<td><strong>Jan 8</strong></td>
<td><strong>Jan 9</strong></td>
<td><strong>Jan 10</strong></td>
<td><strong>Jan 11</strong></td>
</tr>
<tr>
<td>Course</td>
<td>Introduction</td>
<td>Microproc.</td>
<td>Instruction</td>
<td>Set</td>
</tr>
<tr>
<td></td>
<td>LAB INTRO</td>
<td>Overview</td>
<td>LAB INTRO</td>
<td>LAB INTRO</td>
</tr>
<tr>
<td><strong>Jan 14</strong></td>
<td><strong>Jan 15</strong></td>
<td><strong>Jan 16</strong></td>
<td><strong>Jan 17</strong></td>
<td><strong>Jan 18</strong></td>
</tr>
<tr>
<td>Instruction Set</td>
<td>LAB 1</td>
<td>Assembly</td>
<td>LAB 1</td>
<td>Assembly</td>
</tr>
<tr>
<td></td>
<td>LAB 2</td>
<td>Programming</td>
<td>LAB 2</td>
<td>Programming</td>
</tr>
<tr>
<td><strong>Jan 21</strong></td>
<td><strong>Jan 22</strong></td>
<td><strong>Jan 23</strong></td>
<td><strong>Jan 24</strong></td>
<td><strong>Jan 25</strong></td>
</tr>
<tr>
<td>MLK Day</td>
<td>LAB 3</td>
<td>GPIO</td>
<td>LAB 3</td>
<td>GPIO</td>
</tr>
<tr>
<td></td>
<td>LAB 4</td>
<td>Interrupts</td>
<td>LAB 4</td>
<td>Embedded C</td>
</tr>
<tr>
<td><strong>Jan 28</strong></td>
<td><strong>Jan 29</strong></td>
<td><strong>Jan 30</strong></td>
<td><strong>Jan 31</strong></td>
<td><strong>Feb 1</strong></td>
</tr>
<tr>
<td>Interrupts</td>
<td>LAB 5</td>
<td>Embedded C</td>
<td>LAB 5</td>
<td>Embedded Sys Design</td>
</tr>
<tr>
<td><strong>Feb 4</strong></td>
<td><strong>Feb 5</strong></td>
<td><strong>Feb 6</strong></td>
<td><strong>Feb 7</strong></td>
<td><strong>Feb 8</strong></td>
</tr>
<tr>
<td>Embedded C</td>
<td>LAB 6</td>
<td>Timers</td>
<td>LAB 6</td>
<td>Timers</td>
</tr>
<tr>
<td></td>
<td>LAB 7</td>
<td>Embedded Sys Design</td>
<td>LAB 7</td>
<td>Embedded Sys Design</td>
</tr>
<tr>
<td><strong>Feb 11</strong></td>
<td><strong>Feb 12</strong></td>
<td><strong>Feb 13</strong></td>
<td><strong>Feb 14</strong></td>
<td><strong>Feb 15</strong></td>
</tr>
<tr>
<td>Multiplexing</td>
<td>LAB 8</td>
<td>DAC</td>
<td>LAB 8</td>
<td>DAC</td>
</tr>
<tr>
<td><strong>Feb 18</strong></td>
<td><strong>Feb 19</strong></td>
<td><strong>Feb 20</strong></td>
<td><strong>Feb 21</strong></td>
<td><strong>Feb 22</strong></td>
</tr>
<tr>
<td>REVIEW</td>
<td>LAB PRAC 1</td>
<td>DAC</td>
<td>LAB PRAC 1</td>
<td>ADC</td>
</tr>
<tr>
<td></td>
<td>LAB PRAC 1</td>
<td>Mini-Project</td>
<td>LAB PRAC 1</td>
<td>Embedded Sys Design</td>
</tr>
<tr>
<td><strong>Feb 25</strong></td>
<td><strong>Feb 26</strong></td>
<td><strong>Feb 27</strong></td>
<td><strong>Feb 28</strong></td>
<td><strong>Mar 1</strong></td>
</tr>
<tr>
<td>ADC</td>
<td>LAB 6</td>
<td>PWM</td>
<td>LAB 6</td>
<td>PWM</td>
</tr>
<tr>
<td></td>
<td>LAB 7</td>
<td>DMA</td>
<td>LAB 7</td>
<td>DMA</td>
</tr>
<tr>
<td><strong>Mar 4</strong></td>
<td><strong>Mar 5</strong></td>
<td><strong>Mar 6</strong></td>
<td><strong>Mar 7</strong></td>
<td><strong>Mar 8</strong></td>
</tr>
<tr>
<td>DMA</td>
<td>LAB 7</td>
<td>DMA</td>
<td>LAB 7</td>
<td>SPI</td>
</tr>
<tr>
<td><strong>Mar 11</strong></td>
<td><strong>Mar 12</strong></td>
<td><strong>Mar 13</strong></td>
<td><strong>Mar 14</strong></td>
<td><strong>Mar 15</strong></td>
</tr>
<tr>
<td>Spring Break</td>
<td>LAB 8</td>
<td>Spring Break</td>
<td>LAB 8</td>
<td>Spring Break</td>
</tr>
<tr>
<td><strong>Mar 18</strong></td>
<td><strong>Mar 19</strong></td>
<td><strong>Mar 20</strong></td>
<td><strong>Mar 21</strong></td>
<td><strong>Mar 22</strong></td>
</tr>
<tr>
<td>SPI</td>
<td>LAB 9</td>
<td>I2C</td>
<td>LAB 9</td>
<td>I2C</td>
</tr>
<tr>
<td><strong>Mar 25</strong></td>
<td><strong>Mar 26</strong></td>
<td><strong>Mar 27</strong></td>
<td><strong>Mar 28</strong></td>
<td><strong>Mar 29</strong></td>
</tr>
<tr>
<td>UART</td>
<td>LAB 10</td>
<td>UART</td>
<td>LAB 10</td>
<td>Embedded Sys Design</td>
</tr>
<tr>
<td><strong>Apr 1</strong></td>
<td><strong>Apr 2</strong></td>
<td><strong>Apr 3</strong></td>
<td><strong>Apr 4</strong></td>
<td><strong>Apr 5</strong></td>
</tr>
<tr>
<td>Embedded Sys</td>
<td>LAB 11</td>
<td>Embedded Sys</td>
<td>LAB 11</td>
<td>Embedded Sys</td>
</tr>
<tr>
<td>Design</td>
<td>LAB 12</td>
<td>Design</td>
<td>LAB 12</td>
<td>Design</td>
</tr>
<tr>
<td><strong>Feb 22</strong></td>
<td><strong>Feb 23</strong></td>
<td><strong>Mar 1</strong></td>
<td><strong>Apr 24</strong></td>
<td><strong>Mar 24</strong></td>
</tr>
<tr>
<td>&lt;no lecture&gt;*</td>
<td>LAB PRAC 2</td>
<td>LAB PRAC 2</td>
<td>LAB PRAC 2</td>
<td>Embedded Sys Design</td>
</tr>
<tr>
<td></td>
<td>LAB PRAC 2</td>
<td>Embedded Sys</td>
<td>LAB PRAC 2</td>
<td>Embedded Sys</td>
</tr>
<tr>
<td></td>
<td>LAB PRAC 2</td>
<td>Design</td>
<td>LAB PRAC 2</td>
<td>Design</td>
</tr>
<tr>
<td><strong>Apr 25</strong></td>
<td><strong>April 26</strong></td>
<td><strong>Spark Challenge</strong></td>
<td>**Design</td>
<td><strong>Showcase</strong></td>
</tr>
<tr>
<td>ADC</td>
<td>LAB 13</td>
<td>Mini-Project</td>
<td>LAB 13</td>
<td>Mini-Project</td>
</tr>
<tr>
<td><strong>Apr 26</strong></td>
<td><strong>Spark Challenge</strong></td>
<td><strong>Design</strong></td>
<td><strong>Showcase</strong></td>
<td><strong>3:00 - 7:00</strong></td>
</tr>
</tbody>
</table>

*Mini-Project demos/consultation during scheduled lecture hour
The lab for this course is located in room EE 069. You must consistently attend the lab division for which you have officially registered. Quizzes will be given at the beginning of each lab period. You may not use a computer/calculator/phone for this. Pre-lab exercises that are assigned must be finished and submitted by the beginning of your scheduled lab. Steps of experiments must be demonstrated to your lab instructor as they are completed. (No longer need to print lab experiments.) All work for a given lab must be completed and submitted by the end of your scheduled lab period to receive credit. Make-ups require an officially excused absence and pre-approval by your Lab Instructor.
Homework

- Problem sets will be posted on the web site
- Watch for the due dates
- No credit will be awarded for late homework
- Your first assignment is to read the Course Policies & Procedures and Lab Policies and Procedures documents posted on the web site and take a student information survey.
Class Participation

- Bring your iClicker to each class meeting - a properly registered, working iClicker is required to earn class participation credit - no exceptions will be made
- Attendance is required to earn class participation credit - no exceptions will be made
- Register your iClicker on Blackboard – use will begin on Friday, January 11.
- Do not “freak out” if you forget your iClicker or it malfunctions – several sessions will be “dropped”
Office Hours

Scheduled office hours for all course staff members are posted on the course web site.

We will also be using Piazza to facilitate on-line discussion.

Lab Office Hours (Monday-Thursday, 7:00-10:00 PM) will start January 14.
Mini-Project

- Embedded system design based on STM32F051 Discovery Kit
- Basic requirement is to design a product that makes good use of the processor’s computational and interfacing resources
- Done in teams of 2-4 students (self-selected)
  - Unless you don’t team up, then it will be non-self-selected.
- Rick has lots of rules about what you can and cannot do.
How will I apply what I learn in this course?

Sample ECE 362 Mini-Projects and ECE 477 Senior Design Projects
ECE 362 Mini-Project Design Showcase
Do you wish you could be **Frame Perfect**?
A macro controller for the GameCube

**What is a GameCube macro?**
- An automatic sequence of controller inputs that can cause a game to behave in a certain way.
- Macros can provide input faster and more precisely than human hands.
- This may give a player a competitive advantage and can even be used to create effects and combos that are impossible to do with human input.

**How does the GameCube talk to the controller?**
Single line master slave protocol
- 0 bit
- 1 bit

Gamecube asks for data  Controller responds

**How do you start a macro?**
- **“Short Hop Double Laser”**
- **“Moon Walk”**
- **“Waveshine”**
- **“Drill Shine”**
- **“Multishine”**
- Switch between macro schemes
- Macro Button - Hold to enable macros

**How do we intercept the communication?**
1. Gamecube requests data from controller
2. Micro reads from controller, PLD writes previous (possibly altered) data to GameCube*
3. Micro loads PLD with new data

*This adds one cycle (8ms) of delay

**Example of a macro - “Waveshine”**

<table>
<thead>
<tr>
<th>Time (ms)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>no input</td>
</tr>
<tr>
<td>31</td>
<td>🟥 + 🟪</td>
</tr>
<tr>
<td>94</td>
<td>🟪</td>
</tr>
<tr>
<td>109</td>
<td>🟥 + 🟪</td>
</tr>
<tr>
<td>156</td>
<td>no input</td>
</tr>
</tbody>
</table>

[https://www.youtube.com/watch?v=M972kv7pbNk&ab_channel=TheKadiekof](https://www.youtube.com/watch?v=M972kv7pbNk&ab_channel=TheKadiekof)
The Aqualizer

Filter Design

Transfer Functions of Filters

Aqualizer is in action.

Software Design

Electronics

https://www.youtube.com/watch?v=ovuchyM2fDg&feature=youtu.be
The Flascii Bird is a real life simulation of the game flappy bird. The player is asked to wear two specially designed gloves that have accelerometers embedded. The game is entirely controlled by movements of the player and the data collected from the accelerometers is sent wirelessly to the HCS12 microcontroller. The Papilio One FPGA is responsible for outputting VGA graphical interface to the monitor.

The Papilio One is an open-source development platform. It utilities the power of FPGA through implementation VGA driver with SPI communication.

Features:
- 4Mbit SPI Flash Memory
- Two-Channel USB Connection for JTAG and Serial Communications
- 48 I/O lines

Overview

Component List

- HCS12 Microcontroller
- Papilio One FPGA
- Xbee 1mW Trace Antenna Series 1
- ADXL 335 Accelerometer
- Printed Circuit Boards

Papilio One – 500K

PCB Design

Block Diagram

Accelerometer Data

Xbee Transmitter

Xbee Receiver

Low Pass Filter

ADC

HCS12 Microcontroller Unit

Game Logic

TIM

Papilio One FPGA

VGA

PWM

Buzzer

Monitor Output

http://youtu.be/LIovdILAlTs
The purpose of this project is to build a surveillance video camera, whereby the direction of the camera can be controlled by a user using two potentiometers. The camera covers a range from 60 degrees left to 60 degrees right and 60 degrees up to 60 degrees down. The push button then toggles the machine between run mode and hold position mode. The Liquid Crystal Display (LCD) panel will display the angle of the camera and the corresponding Light-Emitting Diode (LED) rows will also light up as an additional indicator. If there is no user input after a whole minute, the machine will return to its initialization stage and the welcome screen will display on the LCD panel.

Microprocessor used: 9S12C32

Peripherals used:
- RTI: The two buttons
- SPI: LCD screen
- ATD: Potentiometers
- TIM: Reset
- PWM: Motors

https://www.youtube.com/watch?v=cHe7elEZAPk
Raulmaha Digital Synthesizer

Andrew Pawling, Aimad Md Uslim, John Sterrett, Chris Pierson

Project Design Goals

The creation of a digital synthesizer that integrates the following features:

- Mono-phonic, single octave keyboard range synthesizer
- Analog volume and distortion control
- Variety of audio waveforms (sine, triangle, saw tooth, and square)
- Accelerometer-based pitch bend ability

https://www.youtube.com/watch?v=148smSUeaGM
Don’t Panic

• The mini-project has gained a reputation of weeks of sleepless nights for its participants. It doesn’t have to be like this.

• When you do a mini-project, you do NOT need to:
  – Produce a polished, commercially-viable result.
  – Disrupt the market.
  – Become a captain of industry.
  – Raise venture capital.

• Find a few people you like working with, chose a project that lets you learn, is meaningful to you, and you can cherish hereafter. Then go talk to your TA and Rick about how to build it. Do so early.

• The most rewarding aspect of all my teaching experiences has been working with mini-project teams and seeing their results.
ECE 477 Senior Design Project

- Open-ended, team-specified embedded microcontroller-based device of personal interest to (most) team members
- Done in teams of 4 students (self-selected)
- Tractable, yet “difficult enough”
- Must utilize a microcontroller and may also utilize a CPLD or FPGA
- May also utilize a “motherboard” (e.g. R-pi)
- Must interface to something: sensor, keypad, LCD, etc. using USB, Ethernet, Firewire, IR, RF, etc.
- Requires the design of a two-layer custom printed circuit board (PCB)
- Must satisfy five general and five project-specific success criteria
- Opportunity to develop communication and teamwork skills that will be needed in industry
das Autötünr

A feat of Purdue engineering, das Autötünr is an automatic guitar tuner and MIDI transcription device. It has a motor assembly of six servo motors which turn the pegs of a guitar based to a default or user-defined tuning. A discrete Fourier transform is used to identify the fundamental frequency of a sound signal. The motors will adjust the strings to match the correct frequency. The frequencies can also be stored as a MIDI file to a USB mass storage device.
Not So Deep Blue is an electronic chess board.

The product enhances game experience by displaying possible moves when a piece is picked up. Piece detection is achieved using reed switches under each square on the board and magnets on each piece. Game information is displayed using RGB LEDs. This includes game time, player turn, piece location and possible moves.
“We Will Find You” is a group of three distributed modules that detect and determine the location of a cell phone in a defined area.

Triangulation Method

- The main module and a pair of satellite modules are placed to the corners of an isosceles triangular area.
- The power levels received by the antennas are transmitted to the main module.
- The location of the cellular phone is calculated by the main microcontroller using triangulation method.
- It is displayed on a web browser through Ethernet connection.

Main and Satellite Modules

Project Specific Success Criteria

- An ability to send data from two satellite antenna modules to the main microcontroller.
- An ability to measure band-specific RF energy generated by cell phones through designed antennas and receivers.
- An ability to determine the position of a cell phone relative to the antenna modules’ positions.
- An ability to render a graphical display of the position of the detected cell phone.
- An ability to calibrate the device for different sized detection areas based on triangular antenna arrangement.
RoboRubik is a self-contained automated Rubik's Cube solver. You can scramble and solve it just like a normal Rubik's Cube. If you get stuck, you can get hints as to what your next move should be. RoboRubik comes with an embedded user interface accessed through any device with wireless networking ability. It's a fun and simple way to learn about and play with one of the most beloved puzzles in the world.

© 2008 RoboRubik
The Two Wheel Deal is a vehicle used for transporting a single rider on two wheels. The design uses an accelerometer and gyroscope to sense when the center of gravity is not directly over the axis of the wheels. It then drives the wheels in order to keep the vehicle balanced. The LCD screen displays battery life, speed, and tilt angle.
Hooked on Harmonix is a learning tool that teaches the user valuable piano skills while providing an entertaining experience at the same time.

**Synopsis:** The user selects one of several MIDI tracks stored in Flash memory, and the corresponding file is parsed outputting bars of appropriate length to a standard computer monitor. Performance from a standard MIDI keyboard is judged and graded when the song is finished.

User input from standard MIDI keyboard is compared to stored MIDI song file.
SOUNDS GOOD / DS³ Digital Steerable Sound System

Joe Land, Ben Fogle, James O’Carroll, Elizabeth Strehlow

PROJECT DESCRIPTION:

- Digitally Steerable Sound System, allows for non-ideal placement of speakers
- Six Preset Equalization Modes
- Wireless Control Interface

ILLUSTRATION OF CONCEPT:

USER MENUS CONCEPT:

PCB LAYOUT:

USER INTERFACE UNIT:

LOUDSPEAKER UNIT:

SIGNAL PATH:
Learning Outcomes

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

- 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 -OR- an average of at least 60% on all lab experiments will be required to demonstrate basic competency.

- Outcome 2 will be assessed based on the midterm lab practical concept exam, for which a score of at least 60% will be required to demonstrate basic competency -OR- an average of at least 60% on labs 1 – 5.

- Outcome 3 will be assessed based on the midterm lab practical concept exam, for which a score of at least 60% will be required to demonstrate basic competency -OR- an average of at least 60% on labs 6 – 10.

- Outcome 4 will be assessed based on the Embedded System Design Mini-Project, for which a score of at least 60% will be required to demonstrate basic competency.
# Grade Determination

<table>
<thead>
<tr>
<th>Percentage Range</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>90% to 100%</td>
<td>A- / A / A+</td>
</tr>
<tr>
<td>80% to 90%</td>
<td>B- / B / B+</td>
</tr>
<tr>
<td>70% to 80%</td>
<td>C- / C / C+</td>
</tr>
<tr>
<td>60% to 70%</td>
<td>D- / D / D+</td>
</tr>
<tr>
<td>&lt; 60%</td>
<td>F</td>
</tr>
</tbody>
</table>

## Bonus Exercises (contingent on Class Participation)

<table>
<thead>
<tr>
<th>Component</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class Participation (iClickers)</td>
<td>4.0%</td>
</tr>
<tr>
<td>Homework Assignments (10 @ 1%)</td>
<td>10.0%</td>
</tr>
<tr>
<td>Lab Experiments (10 @ 2%)</td>
<td>20.0%</td>
</tr>
<tr>
<td>Lab Quizzes (10 @ 0.5%)</td>
<td>5.0%</td>
</tr>
<tr>
<td>Lab Practical Concept Assessment Exams (2 @ 10%)</td>
<td>20.0%</td>
</tr>
<tr>
<td>Lab Practical Programming Assessment Exams (2 @ 13%)</td>
<td>26.0%</td>
</tr>
<tr>
<td>Embedded System Design Mini-Project (Outcome 4)</td>
<td>15.0%</td>
</tr>
</tbody>
</table>

| Total Percentage                                   | 100% + Δ% |
Grade Determination (historical)

- **Note:** There are **no** A / B / C / D / F “quotas”!
- **Goal:** **Minimize** number of D / W / F grades!!

Typical grade distribution and course GPA: 2.92
Borderline Cases and Incompletes

- A “borderline” is officially defined as an NWP within 0.5% of a cutoff.
- Before course grades are assigned, the instructor will carefully examine all such cases to determine if the next higher grade is warranted.
- IMPORTANT NOTE: The “next higher grade” is NOT AUTOMATICALLY GUARANTEED!!
- 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.
- University Regulations stipulate that a student must be PASSING in order to qualify for a grade of I or E.
Academic Integrity

- For purposes of this class, we broadly define academic dishonesty as being any attempt by a student to improve a grade beyond his or her own *personal* understanding of the material in question.
- See the COURSE POLICIES AND PROCEDURES document for a longer discussion.
- You will be faced with choices. Call them moral dilemmas.
- Keep in mind we use computer software to find similarities between students’ work. Inordinately similar work is an indication of a lack of personal understanding, and it has penalties...
The minimum penalty for academic dishonesty is a zero on the assignment in question and a single letter drop for a final course grade.

All cases of academic dishonesty will be reported to the ECE Associate Head, the ECE Undergraduate Counseling Office, and Office of Student Rights and Responsibilities.

Activities that are considered to be dishonesty are listed in the COURSE POLICIES AND PROCEDURES document.

Unless explicitly stated, we expect that no student will work with any other student on work to be submitted for a grade.
Emergency Preparedness

- To report an emergency, call 911
- To obtain updates regarding an ongoing emergency, sign up for Purdue Alert text messages, or view current status at [www.purdue.edu/ea](http://www.purdue.edu/ea)
- There are nearly 300 Emergency Telephones outdoors across campus and in parking garages that connect directly to the PUPD – if you feel threatened or need help, push the button and you will be connected immediately
- If a fire alarm sounds during class we will immediately suspend class, evacuate the building, and proceed outdoors – do not use the elevator
- If we are notified during class of a Shelter in Place requirement for a tornado warning, we will suspend class and shelter as directed
- If we are notified during class of a Shelter in Place requirement for a hazardous materials release or a civil disturbance (including a shooting or other use of weapons), we will suspend class and shelter in the classroom, shutting/securing the door and turning off the lights
Important Deadlines/Restrictions

- **All lab division changes** must be done through [MyPurdue](https://mypurdue.purdue.edu) during the first week of classes.
- You must attend the lab division for which you have officially registered.
- **No late pre-labs or homework** will be accepted.
- Requests for **make-up labs** must be approved by your Lab Instructor in advance of the evening office hour session you plan to complete the makeup.
- Quizzes will be given at the **beginning** of your scheduled lab period – **there will be no make-ups** (quizzes missed due to officially approved absences will be **pro-rated** – maximum is 2).
- **Makeup exams** for planned absences on scheduled exam dates must be arranged in advance and completed during the week the exam is being administered.
- Exams missed due to **officially excused absences** (illness or family emergency) will be **made up during finals week**.
Let’s Get Started!

- The first part of this course will cover assembly language and C programming techniques.
- The second part will cover microcontroller interface design techniques.
- The third part will cover microcontroller peripherals and application examples.
- The fourth part will cover embedded system design considerations.
- *All* of the topics covered in this course have been *carefully chosen*, based on how the material will be used in later courses.