ECSE-2350
Embedded Control
Lecture 1

Course Introduction
C Variables and Operations

Outline

- Embedded Control
  - What is it?
  - Where is it?
- Microprocessors and Microcontrollers
- Course Policies and Format

- Number Systems

Embedded Systems

...the things this class is about.

An Embedded System is

- Any system or device that is capable of computation and control tasks without **significant** user intervention.
  - These systems USUALLY have at least three components of:

1) **Control Outputs** (motors, lights, comms, etc.)
2) **Sensor Inputs** (accelerometers, encoders, etc.)
3) **Processing** (microcontroller, microprocessor, FPGA, etc.)
4) **User Interface** (buttons, screen, audio, etc.)
5) **Data Management** (storage, wireless communications, etc.)

Examples


Visa Signature
4000 1234 5678 9010
4000
GOOD THRU 12/20
A. MILLER
VISA
Signature

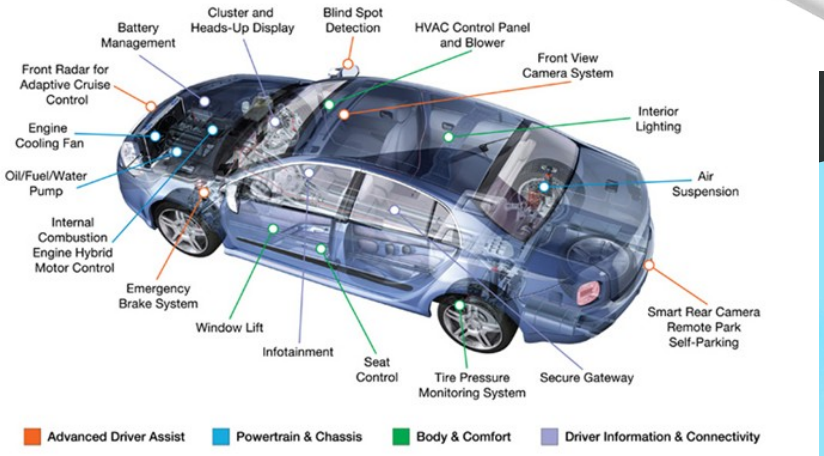

E-ZPass

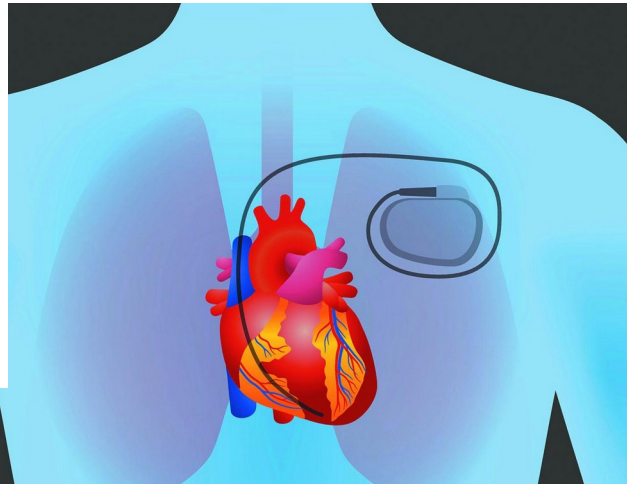

Rakuten


reolink


TCL · Roku TV
12:40 pm | Options
Home
My Feed
Movie Store
TV Store
News
Search
Streaming Channels
Settings
Cable
Antenna
Game console
NETFLIX
hulu
YouTube
HBO NOW
ESPN
pandora


nest
COOLING
75
78


Battery Management
Cluster and Heads-Up Display
Blind Spot Detection
HVAC Control Panel and Blower
Front Radar for Adaptive Cruise Control
Front View Camera System
Engine Cooling Fan
Interior Lighting
Oil/Fuel/Water Pump
Air Suspension
Internal Combustion Engine Hybrid Motor Control
Emergency Brake System
Smart Rear Camera Remote Park Self-Parking
Window Lift
Infotainment
Seat Control
Tire Pressure Monitoring System
Secure Gateway
Advanced Driver Assist
Powertrain & Chassis
Body & Comfort
Driver Information & Connectivity



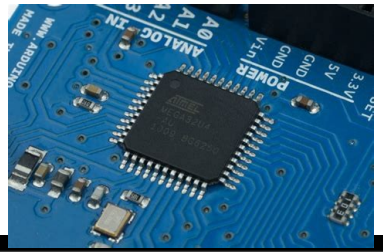
We will start with the Processing...

- Microprocessor
  - Single chip
  - Contains CPU, MMU, …
  - Handles and processes instructions
  - Limited memory, no clocking, no direct I/O
  - Useless by itself

vs.

- Microcontroller
  - Similar to Microprocessor
  - Has more memory, clocking, direct I/O
  - Lots of other stuff “Peripherals”
    - Timing, ADC, DAC, Comms, GPIO, LCD, Decoders,…
  - Basically a computer on a chip!
  - Cheap.

AMD Ryzen 9 7950X3D



Atmel MEGA32
(Arduino LEONARDO)

We will use the MSPM0G3507

- New last semester...
- ARM Cortex-M0+ (very low-power, simple)
- Max 80 MHz clock –
  - we'll be running at 32 MHz
- 128 kB FLASH (program memory)
- 32 kB RAM (working memory)
- A bunch of other stuff →

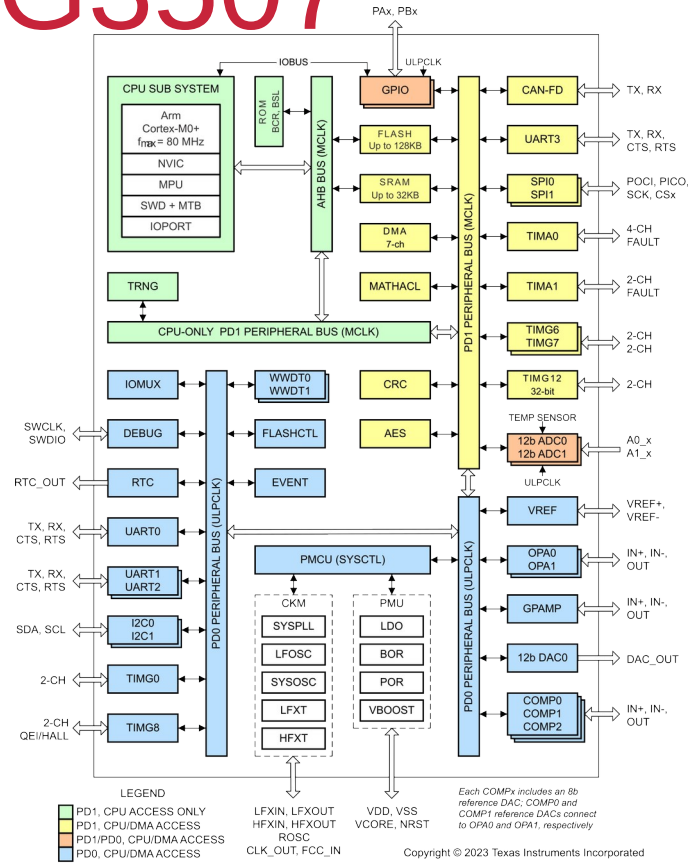
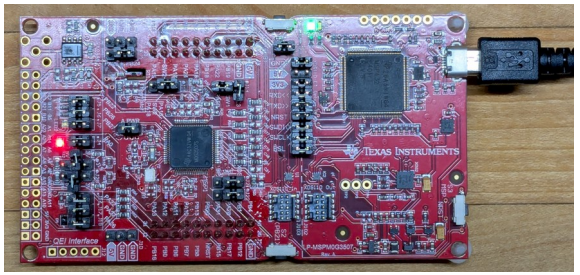
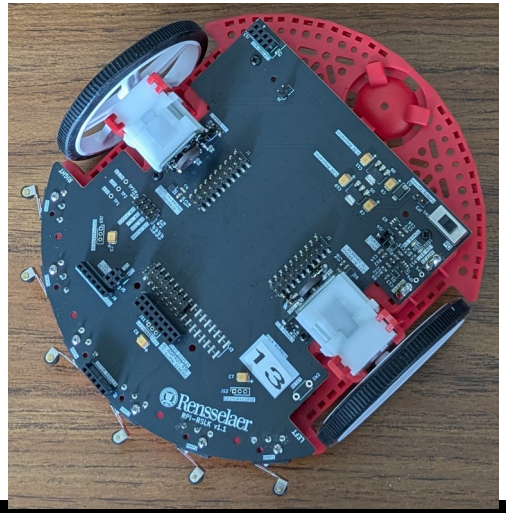


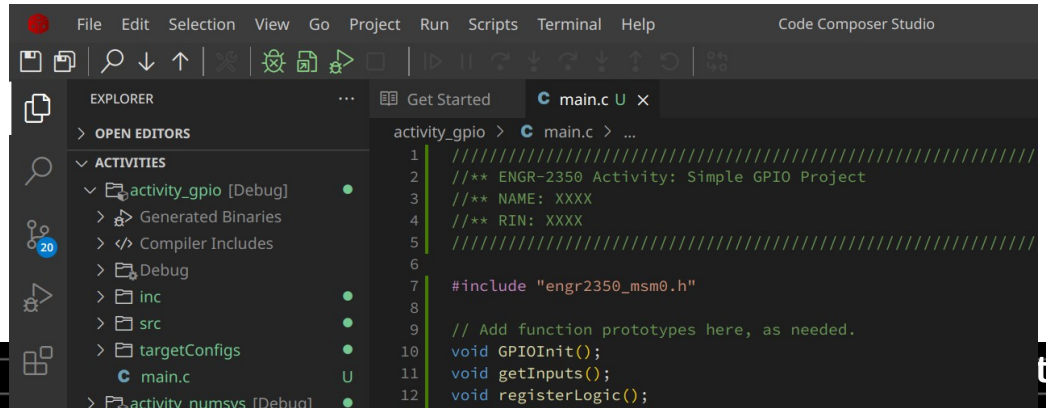
Figure 4-1. MSPM0G350x Functional Block Diagram

Working with the MSPM0G3507...

- Hardware:
  - LP-MSPM0G3507 “Launchpad” development board
  - RPI-RSLK Robotic Car
- Software:
  - Ti Code Composer Studio







What We'll Do...

- Learn the basics of building of building fused hardware/software, "embedded" systems:
  - Adding additional circuits to the microcontroller
  - Both reading and controlling said circuits
  - Building functional software (or "firmware") in C
  - Learn basic functionality of common microcontroller peripherals
    - GPIO, Timers, Analog Subsystems, Serial Communication, etc.
  - Improve upon debugging and troubleshooting skills

Course Information and Syllabus

...extra credit if you don’t fall asleep

Staff

- Instructors: Kyle Wilt (S2), Prabhakar Neti (S1)
  - Office Hours:
    - All scheduled class meeting times
    - Others will be announced (see website)
- Grad TAs: Each section will have one or two TAs
  - The TAs will do the bulk of grading.
- A handful of UG-SAs for helping out ← they know what's going on very well!

Course Tools

- **Course Website:** sites.ecse.rpi.edu/courses/static/ENGR-2350/
- **Webex:** Group space for discussion as needed.
- **Gradescope:** All assignments submitted here.
- **LMS:** Hub and announcements.

Calendar

- Weekly Schedule:
  - Laboratory Sessions on M/Th, **JEC2104**
    - Used primarily for working on Activities/Homeworks and Laboratories.
    - Will have a few lectures during this time.
    - Attendance may be taken randomly throughout semester.
  - Lecture Sessions on Wednesday Morning (10:00 AM – 11:50 AM) **HERE**
    - Exams will be given during this time slot as well.
    - Attendance is not “required” for Lectures but strongly preferred.
      - This might change if we notice low turnout...

Calendar

- Important Dates:
  - **1/19 Martin Luther King Jr. Day:** No class.
  - **2/16 President's Day**: Monday classes on Tuesday 2/17.
  - **3/2-3/6 Spring Break**
  - **3/23-3/27 GM Week.** No Lecture on Wednesday.
  - Exam 1: February 25th Exam 2: April 22nd No Final
  - Quizzes (3x): See Calendar

Gradeables

- 2x Exams, 2 hr. Closed book/notes/computer
  - We will be provide a crib sheet, shared in advance
- 3x Quizzes: ½ hr. Open book/notes/computer
- 14x Activities/Homeworks
  - Assigned during lab session, due after next class
- 5 Labs and 1 Final Project*

*Depends on how semester goes, might be lab extension

Class Format

- [14] Activities (+Homeworks) will be assigned for new topics or for new C skill practice.
  - In-person work (usually)
  - Online questions (sometimes)
  - Coding portion (usually)
  - Lowest score dropped
  - Late Activities accepted at -50%
  - Activities may require ADALM2000 or Analog Discovery.
    - Will need for future classes.

Component	Quantity	Weight
Exams	2	40 %
Quizzes	3	12 %
Activities	14[1]	20 %
Laboratories	5	15 %
Final Project	1	8 %
Attendance and Participation	-	5 %
	Total:	100 %

Class Format

- [3] Quizzes will be given (see Calendar)
  - ~½ hour per quiz
  - Given during LAB TIME
  - Will discuss as they approach.
  - Open Open Book/Open Notes
    Open Computer – No (Human/AI) Chats!

Component	Quantity	Weight
Exams	2	40 %
Quizzes	3	12 %
Activities	14[1]	20 %
Laboratories	5	15 %
Final Project	1	8 %
Attendance and Participation	-	5 %
	Total:	100 %

Class Format

- [2] Exams will be given on paper during lectures
  - Closed Book/Closed Notes
  - Single Crib Sheet Provided
  - You may have input on Crib Sheet.*

*but cannot bring your own

Component	Quantity	Weight
Exams	2	40 %
Quizzes	3	12 %
Activities	14[1]	20 %
Laboratories	5	15 %
Final Project	1	8 %
Attendance and Participation	-	5 %
	Total:	100 %

Class Format

- 5 Laboratories and a Final Project – Need to do all **Labs** successfully*!!
  - Using content from lectures, develop an embedded system to perform certain tasks:

1) General I/O (+GPIO)
2) Game (+Timers, Interrupts)
3) Prescribed Driving (+PWM+Encoders)
4) Cruise Control (+P Control)
5) Informed Driving (+Serial Sensors)

Final Project (All of it-ish)

Component	Quantity	Weight
Exams	2	40 %
Quizzes	3	12 %
Activities	14[1]	20 %
Laboratories	5	15 %
Final Project	1	8 %
Attendance and Participation	-	5 %
	Total:	100 %

* Not submitting all labs successfully will incur a significant grade penalty or failure. See Syllabus for more details.

Expectations

- We do not expect you to be able to do the labs completely on your own, though we will try our best to prepare you for them.
  - We are here to help – Reach out early and often!
  - We will not do the labs for you however, it should be obvious to us that you've thought about the problem.
  - **Do NOT be shy asking questions. Do you have a question? Having problems with your lab? ASK IT!!!**
    - **Do It.**

How to get help...

- Access to attending staff during Labs
- Open Shop hours (To be posted)
- Post a question Webex on off hours – hopefully someone will get back to you eventually
- Office hours: See course website
- I prefer Webex chats!

Submitting work...

- Everything will be going to Gradescope
  - If assignment doesn't exist or the date is wrong, let us know! It is easy to make an error on creating these things.
- Everything should be YOUR OWN WORK (or your group's own work).
  - Changing variable names or comments or spacing from someone else's code does not constitute "your own work"
  - We use the Gradescope code similarity tool to check!
  - Snippets of code copied from elsewhere (for example: ChatGPT, Sourceforge) ***must be attributed*.**
- Text/Number Entry in Gradescope:
  - Make sure you enter your answer EXACTLY as specified by the problem. Many of these problems are autograded and will not be marked correct when not exactly equal.
    - If you answered correctly but it doesn't match exactly, submit a regrade and we'll either fix or give partial credit.

Submitting work...

- Coding Activities:
  - Submit main code file only (edited *.c file)
  - No late submissions, minus pre-approval or official Dean's excuse.
  - **Must compile to get *any* points**, regardless of the correctness of the rest of the code.
    - That is, it must be clear you compiled and tested it.

Submitting work...

- Labs:
  - Labs graded out of 20.
  - Checkoffs: Labs have two “parts”, where each part must be checked off before the next class, as specified by the calendar
    - Part A checkoffs “Intermediate checkoffs” **will not be accepted after the due date**.
      - **Worth 5 points of total Lab grade**
    - Part B checkoffs “Final checkoffs” will be accepted late: 25% deduction for each week late
      - **Worth 15 points of total Lab grade.**
  - Must submit complete and working code after Final Checkoff to receive lab credit.
    - Late submission of code will not necessarily invoke late checkoff penalty

You should do well if ...

- You do not procrastinate!
  - The above bullet point is the main reason students have difficulty in this class.
- **If you’re stuck on something, you ask us! Don’t fight with the same problem for hours and hours only to get more and more frustrated.**

C Variables and Operations