EECS 192: Mechatronics Design Lab

Discussion 1: Introduction

GSI: Richard ”Ducky” Lin 🦆

21 Jan 2015 (Week 1)

- Administrivia
- FRDM Board Intro
- Soldering
Welcome to EE192!
Project

- Project: build an autonomous track-following racecar given a stock chassis and microcontroller dev kit
- Teams should be 3 students
  - Combined skillset should include mechanical design / fabrication, electronics, programming
  - Controls experience helpful
- Teams formed by checkoff Friday
- Read the competition rules
  - Freescale Cup
  - NATCAR
Checkoffs

- One-hour time slot on Friday 11:30am-12:30pm to demonstrate that your project is where it should be
- At least one team member needs to show up to run your hardware
- These are graded, half credit if late

- First checkoff this Friday
  - Form project teams and check out cars
  - Checks4Cars program: trade a $300 deposit check for a car
  - Get private course GitHub repository
  - Details on website

Get your cars!
Git Refresher

- Git: distributed version control software
  - Each commit: like complete snapshot
  - Branches: separate chains of commits
    - eventually merged back to its parent
  - Distributed: everyone has complete copy
    - Most operations local, periodically sync

- Best Practices
  - Small, logical, often commits
  - Write good commit messages
  - Develop in branches: keep master clean

Learn git here: try.github.io

git logo, by Jason Long, CC BY 3.0
FRDM-KL25Z Development Board
MKL25Z128VLK4 microcontroller
  ▶ 48MHz ARM Cortex-M0+
  ▶ 128KB flash
  ▶ 16KB SRAM
Programmable using USB
I/O headers including
  ▶ GPIO
  ▶ 16-bit analog inputs (ADC)
  ▶ 12-bit analog output (DAC)
  ▶ PWM, I²C, SPI, and UART modules
On-board RGB LED and accelerometer
IO Refresher

- **GPIO (general purpose input/output) pins**
  - As an output: sets voltage on pin from software, either GND (0) or Vdd (1)
  - As an input: samples voltage on the pin, returning either 0 (low) or 1 (high)

- **PWM (pulse-width modulation) module**
  - Every *period*, the pin is high based on the *duty cycle*, then low for the remainder
  - Can digitally approximate analog outputs

- **Analog Inputs (ADC)**
  - Converts a continuous analog voltage (0-3.3v) to a 16-bit (0-65535) quantity
Concurrency Refresher

- FRDM-KL25Z’s processor is single core
- Blocking Operations
  - Operations do not return until finished, blocking thread of control
  - IO operations may be lengthy!
- Nonblocking Operations
  - Operations return immediately, activity continues in the “background”
  - IO operations can buffer data and use interrupts to send/receive data
- Threading and RTOS
  http://developer.mbed.org/handbook/RTOS
  - mBed has a RTOS with threading, concurrency, and synchronization
  - Beware of threading anti-patterns
MODSERIAL serial(USBTX, USBRX);

DigitalOut led_green(LED_GREEN);
DigitalOut led_red(LED_RED);
PwmOut led_blue(LED_BLUE);

int main() {
    // Internal LED is active low.
    led_green = 0;
    wait(0.25);
    led_green = 1;
    wait(0.25);

    // Mandatory "Hello, world!"
    serial.printf("Hello,World!\r\n");

    // Run led_fade_thread() in own thread
    Thread ledFadeThread(led_fade_thread);

    // Periodically call led_blink_periodic()
    RtosTimer ledBlinkTimer(led_blink_periodic);
    ledBlinkTimer.start(1000);

    // Work is done in the threads,
    // so main() can sleep.
    Thread::wait(osWaitForever);
}

void led_fade_thread(
    void const *args) {
    // Note this doesn't terminate.
    while (1) {
        // Invert duty cycle.
        led_blue.write(1-0);
        Thread::wait(250);
        led_blue.write(1-0.25);
        Thread::wait(250);
        led_blue.write(1-0.5);
        Thread::wait(250);
        led_blue.write(1-0.75);
        Thread::wait(250);
    }
}

void led_blink_periodic(
    void const *args) {
    // Toggle the red LED when called.
    led_red = !led_red;
}
Live Demo!

This is essentially the procedure demonstrated in the checkpoint 1 page

... and hopefully goes Murphy-free ...

Note: you’ll have to download the Device Family Pack for the FRDM-KL25Z
http://www.keil.com/dd2/arm/armcm0/
(also on the checkpoint page)
Soldering: joining (electrically and mechanically) metals using a separate filler metal “solder”

Electronics: bonding component pins/leads to circuit board through-holes or pads
  - Solder is usually a tin/lead alloy (e.g. 63/37) or lead-free tin-silver-copper alloy (e.g. SAC305)

This tutorial focuses on introductory through-hole soldering
  - Note: most production boards today are surface-mount to save space
Safety Precautions

- Soldering melts metal - IT’S HOT
  - Tips typically set at 700°F (371°C)
  - Irons can stay hot after turning off
  - Touching a hot tip is NOT fun
- Leaded solder contains, well, lead...
  - ... which is known to the state of California to cause cancer and reproductive harm ...
  - WASH YOUR HANDS AFTERWARDS
- Solder vaporizes flux, producing fumes
  - Regular exposure linked to asthma
  - DON’T BREATHE THEM IN
  - May also cause solder splatter: safety goggles recommended

Lead poisoning: not as fun in real life

© Fox
Soldering depends on good thermal transfer from tip to solder / component / board.

- Metals oxidize, forming an oxide layer
  - Oxides impede thermal transfer
  - Reactions faster at higher temperatures

- Flux provides chemical cleaning
  - Rosin flux is corrosive when heated
  - ... and is present in solder wire spools
  - ... but is “burned” upon use

- Just keep this in mind...
Equipment Overview

Soldering Iron Base
(front view)
- Temperature Adjust Knob
- Wire to Handpiece

(Side view)
- Power Switch

Soldering Iron Handpiece
- Wire to Base

Tip (cutaway view)
- Solder (tinned coating)
- Iron Plate
- Copper Filling

Caution: These parts get very hot during operation! Do NOT touch until cool!
Tip Maintenance

- The tip is what heats things up
  - Want to maximize thermal transfer!
- Keep the tip “tinned” with solder
  - Provides better thermal transfer
  - Sacrificial layer preventing tip oxidation, which destroys the tip
- Must be occasionally refreshed
  - The solder oxidizes, accelerated by heat
  - Cleaning: wipe on brass or wet sponge
  - Immediately re-tin (apply solder layer)
Procedure

- Beginner’s tip: use iron to heat up component and board, not solder
  - Feed solder in through the other side
  - Solder only melts when component and board sufficiently hot
- Maximizing heat transfer
  - Point tips: solder using “side” of tip, not point
  - Chisel tips: use the broad flat end
Joint Inspection

Optimal joint shape is a “solder volcano”
Through-Hole Soldering Demo

Live Demo!

... which REALLY hopefully goes Murphy-free ...
Quick poll: best time for GSI office hours? (about 2 per week)
   - Thursday, for the pre-checkoff scramble?
   - Other times?

Thursday section only: has schedules cleared up enough to move discussion to Wednesday?
   - Otherwise, future discussion sections (starting Thu, 29 Jan) will be 9:30am-10:30am
Electrostatic Discharge

- You build up static charge on your body
  - ... just by walking, especially when it’s dry
  - ... and up to several kV
  - but under $\sim 2\text{kV}$ is imperceptible
- Chips are sensitive to high voltages:
  **may cause permanent damage**
  - read: board stops working “for no reason”
- Remember to ground (discharge) yourself before handling sensitive electronics
  - Touch the grounded lab bench surface
  - Use a ESD wriststrap
  - Avoid touching traces on boards
Get your parts and get started!

I'll be walking around helping!

For checkpoint 1, you need to solder a resistor and LED onto perfboard.
Choose the resistor such that \( \sim 1.6\text{mA} \) goes through the LED.
The MCU supply voltage is 3.3V.

(Yes, I know those red LEDs suck)

Also, grab a computer account form!