EECS 192: Mechatronics Design Lab

Discussion 1: Introduction

GSI: Justin Yim

23 Jan 2019 (Week 1)

- Administrivia
- BeagleBone Blue 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
  - NATCAR
Checkoffs

- One-hour time slot on Friday TBD 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 a complete snapshot
  - Branches: separate chains of commits
    - Eventually merged back to its parent
  - Distributed: everyone has a 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
FRDM-K64F Development Board
Beaglebone Blue (BBBL)

Octavo Systems OSD3358 Processor
- 120 MHz M4 1 GHz ARM Cortex-A8
- 1024 KB 4 GB 8-bit eMMC flash storage
- 256 KB 512 MB RAM

Program over network via USB or Wifi

I/O connectors including
- GPIO
- 12-bit SAR ADC
- USB2.0, I²C, SPI, and UART
- PWM, Servos, Encoders, H-bridges

LEDs, Buttons, 9-axis IMU, Barometer
GPIO (general purpose input/output) pins (strawsondesign.com GPIO link)
- Output: sets pin voltage from software: either GND (0) or Vdd (1)
- Input: samples pin voltage: 0 (low) or 1 (high)

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

Analog Inputs (ADC) (strawsondesign.com ADC link)
- Converts a continuous analog voltage (0-1.8v) to a 12-bit (0-4095) quantity
Robot Control Library

- C library hardware interface for Beaglebone Blue
- Examples and testing programs available
- Documentation: http://strawsondesign.com/docs/librobotcontrol/index.html
```c
#include <stdio.h>
#include <robotcontrol.h>

int main()
{
    if(rc_kill_existing_process(2.0)<-2)
        return -1;
    if(rc_enable_signal_handler()==-1){
        fprintf(stderr,"ERROR...\n");
        return -1;
    }
    rc_make_pid_file();

    printf("Hello World!\n");
    rc_set_state(RUNNING);
    rc_led_set(RC_LED_GREEN, 1);
    int toggle = 1;
    while(rc_get_state()!=EXITING){
        toggle = !toggle;
        rc_led_set(RC_LED_GREEN, toggle);
        rc_usleep(100000);
    }

    rc_led_set(RC_LED_GREEN, 0);
    rc_led_cleanup();
    rc_remove_pid_file();
    return 0;
}
```
Using the Beaglebone Blue

- Accepts power over USB or 9-18V barrel connector
- ALWAYS shut down by either:
  - `sudo poweroff` from SSH
  - "POW" button on board
- Beware short circuits and Electrostatic Discharge (ESD)
- We can only provide you with one replacement board if yours dies
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 \( \sim 2kV \) 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

Don’t let this happen to you
Connecting to Beaglebone Blue

- Connect to USB and wait for boot
- Lab desktops with Windows:
  - Start Virtualbox
  - Pass Beaglebone USB to Ubuntu
- `ssh ubuntu@192.168.7.2`
- Passphrase: temppwd
Hello, World! Demo

Live Demo!

Connect to Beaglebone Blue, Print "Hello World"
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 cool slowly after turning off
  » Touching a hot tip is NOT fun

» Leaded solder contains ... 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 THIS
  » May also cause solder splatter: eyewear required (regular glasses ok)
Oxidation

- 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
  - 250°C
  - 350°C
  - 450°C
- Wire to Handpiece

**Soldering Iron Handpiece**
- Wire to Base
- Grip (hold iron here)
- Barrel (contains heater)
- Tip (melts solder)

**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”
For checkpoints 1 and 2, you need to build 2 LED blink circuits - the first using an IC and the second using PWM from the Beaglebone Blue. Choose the resistor such that $\sim 1.6\text{mA}$ goes through the LED.

The BBBL supply voltage is 3.3v

(Yes, I know those red LEDs suck)
Links

- **Chassis:**
- **Github:** https://github.com
- **Beaglebone Blue:** https://beagleboard.org/blue
- **Robotic Control Library:** http://strawsondesign.com/docs/librobotcontrol/index.html