ECE471: Embedded Systems – Final Project

Due: Friday, 8 December 2017 (Last day of Classes)

Overview:

- Design an embedded system that does something interesting. This is very open-ended, but some guidelines are below.

Guidelines:

- You may work either alone or in groups of two or three. If you work in a group your end project will have higher expectations.
- You may use any embedded board or microcontroller for this project. I may not be able to provide a full amount of help though if you use something other than a Raspberry Pi.
- You may use any programming language you like, but again if it's not in C or Assembler I might not be able to provide a full range of help.
- You may use code/libraries found online as long as licensing allows it and you properly document it. There must be *some* original code done by your group. You cannot just stitch together code found online.
- Your board will have to take input from a user, and display some manner of output. Both of these need to go through one of the low-level hardware interfaces discussed in class (i2c, SPI, 1-wire, GPIO, USB, A/D, PWM, audio in/out, HDMI, etc.)
- Your project may *complement* one being done in another class. However your project must be a new implementation, you cannot just turn in previously done work.

Part 1: Topic Selection (due 6 November 2017) (5pts)

Each group should send a brief e-mail describing your project topic and listing group members.

Part 2: Progress Report (due 20 November 2017) (10pts)

A brief status update detailing progress your group has made. This is primarily to make sure your project is on track to be finished in time; if things are not going well the topic can be adjusted.

Send this report by e-mail. Only one submission is needed per group.

1. State in one sentence a summary of your project.
2. Describe the hardware that you will be using: the embedded board, the input device, and the output device.
3. Have you acquired and tested the hardware mentioned? Are you on track for being finished on time?
4. Will you be willing to volunteer to present early (Monday or Wednesday) rather than the last day?
5. You can submit the status update by e-mail.

Part 3: In-Class Presentation 4, 6 & 8 December 2017 (40pts)

- You will have 10 minutes to present. Plan for 8 minutes of showing off the device and presenting plus 2 minutes for questions. Points will be taken off for going over.
- You may present slides using the projector if you want, but that's not strictly necessary.
- Your presentation should have at least the following information. Feel free to include more.
  - Brief overview of what your device does and how it works.
  - A summary of the hardware being used, including the embedded board
  - Describe the input hardware and how you connect to it
  - Describe the output hardware and how you connect to it
  - A summary of the operating system (if applicable) and the programming language you used, and why.
  - Hardware limits: describe any power concerns
  - Software limits: describe any real time constraints, security concerns, and code density concerns
  - Challenges: list any challenges you had getting things working.
  - Future work: things you might add if you had more time.
  - Leave time to do a brief demo of the hardware.

Part 4: Project Writeup, Officially due 8 December 2017 (45pts)

This will be a short paper (at least 6 pages, but you can include pictures, diagrams, etc.) that must contain all of the following:

1. Introduction: What the device is and high level overview of what it does. Also be sure to make clear what is actually working in your implementation (as opposed to things you wanted to get work but for various reasons did not).
2. Hardware
   (a) Embedded Board Description: Describe the hardware, CPU (architecture, type, speed), RAM, and I/O. Also describe the operating system or other software (kernel version, etc.)
   (b) Input device description: Describe the device you are interfacing with, how you access it in software, and document the protocol you use to communicate with it.
   (c) Output Device description: same as for the input device.
   (d) Links to any data sheets for hardware you used, as well as schematics for any circuits you designed yourself.
   (e) Power Consumption: Explain any energy or power concerns with your application, and how you could optimize it to use less power.
3. Software

(a) Programming Language: Which one did you use? Why? Briefly explain the tradeoffs between the language you chose and doing the same in assembly language. (If your project is in assembly language, the explain the tradeoffs versus C).

(b) Real Time: Does your device have real time constraints? What would happen if your code encountered an unexpectedly large delay?

(c) Security: Describe any computer security issues there might be with your device (can it be exploited?) If you say there are no security issues, make sure you explain why.

(d) If you use code not written by your group (code found online, libraries, etc) explain what the extra code does, and how your code interfaces with it. Explain how much of the code is original to your group.

4. Related Work

   (a) Has anyone done a project like this before?

   (b) How does your project compare to existing similar projects?

5. Conclusion

   (a) If you worked in a group: List who worked on what part.

   (b) Challenges: List any challenges you had, and if things didn't work, explain why.

   (c) Future Work: List any improvements you might make if you had more time and resources to work on the project.

6. Appendix

   (a) The source code (this can be submitted as a separate file, does not have to be included in the report).

   (b) **OPTIONAL** Make a short web-site or YouTube video describing your project. Get it posted on an embedded projects website (`hackaday.com` or similar). No extra points for this, just bragging rights.

   (c) I plan to put a summary of the projects on the course website, possibly including project reports. If you do not want your project posted, please indicate this in the final writeup.

You can e-mail your final report to me. pdf or word document is fine, the code should be attached too.

Hardware Ideas:

- Displays:
  - The LED display from class
  - i2c 8x8 LED grid (some available)
  - i2c LED bargraph (1 available)
  - i2c 4-character alphanumeric display (1 available)
  - 8x16 LCD display (1 available)
  - 8x16 Organic LED display (1 available)

  - 16x2 RGB LCD display (1 available)
  - spi 0.96" 96x64 color OLED display
  - NOKIA LCD display
  - i2c 128x32 monochrome OLED display
- Sensors
  - i2c Temp/Altitude sensor (1 available)
  - i2c pressure sensor
  - i2c accelerometer
  - i2c Light sensor (1 available)
  - i2c color sensor (1 available)
  - i2c UV SI1145 UV sensor
  - A/D distance sensor (1 available)
  - i2c TMP006 infrared temperature sensor
  - i2c INA219 high side current sensor
  - i2c HMC5683L triple-axis magnetometer (compass)
  - 1-wire DS2413 two gpio adapter
  - i2c VCNL4010 proximity/light sensor
  - serial fw5632 GPS receiver
  - serial USB power gauge
- Storage / I/O
  - 1-wire DS2413 two GPIO expander
  - SD card breakout
- Embedded boards
  - Trinket 5V 16MHz
  - Trinket 5V
- Custom interfaces
  - OBDII (car telemetry) to Bluetooth adapter
  - PS/2 keyboard adapter
- Other hardware you can obtain on your own. Some ideas:
  - USB keyboard/mouse/disk
  - Interface to SD card over SPI?
  - Devices hooked to GPIOs (LEDs ? Motors?)

  - USB or rasp-pi web-cam
  - i2c Wii Nunchuck (I have breadboard adapters, you'd have to provide your own nunchuck)
  - Bluetooth devices (will need bluetooth dongle)

Project Ideas:

- Alarm Clock: set time with buttons, play wakeup sound/music over audio out
- Some manner of robot.
- Wii Nunchuck (i2c accelerometer). Show orientation on LED display? Make a simple game? Log acceleration to disk?
- Wii controller to Pi3 via bluetooth?
- Weather/Temperature Display that remembers high/low temperatures
- Audio in on the sound input driving some sort of audio visualization on LED display
- Some sort of video game utilizing LED display
- Using wireless or Bluetooth in an interesting way?
- Color sensing candy sorter
- Car or bike computer
- Hooking old PS/2 style keyboard to Pi using GPIO interface
- Measure the power consumption of Pi doing various things, optimize for this.