EE 285 FINAL PROJECT

ELECTRICAL AND COMPUTER ENGINEERING
IOWA STATE UNIVERSITY

FINAL Project

Objectives

The main objective of the final project is to teach you how to put together all of the class material that you have learned so far in order to create a larger C program that accomplishes a specific task. The goal is to show how programs are designed in the real world.

Project Selection

Feel free to choose any one of the following three projects. They are all of approximately the same difficulty. Your grade will not depend on the project that you pick, as long as you implement the entire project.

If you don't like any of the provided projects, then you also have the option of specifying your own project. If there was something that you always wanted to do with C, now is your chance. However, your proposed project MUST be of appropriate complexity as determined by your lab TA. The minimum constraints are that y*our project MUST: 1) include pointers 2) include functions, and 3) a custom library, a separate source/header file.*

Notes on the Two Provided Projects

The two projects descriptions outline all necessary details for the projects. If you choose to implement one of these projects, **you must follow** all details listed in the project description. Your implementation, at a minimum, **should satisfy** all implementation features and constraints as specified in this document. This means that if you do not implement a certain feature or reduce the complexity of the project you will lose points as indicated in the grading **RUBRIC**. You are allowed to work beyond the basic functionality as long as the implementation improves upon the specified design. It would be helpful to discuss any such changes with your TA.

Think about the final project as a really long homework. You may have to go to the lab outside of your regularly scheduled lab time in order to complete it. To get a grade for this assignment you need to present your final project to your TA during dead week (in your regular lab time). No presentation, no grade!

Also, there will be ZERO credit for any design step in the grading rubric that does not work or produces the wrong output or is not of the given specifications. All components of your design MUST WORK and produce the output as described in the specs to receive credit.

EE 285 FINAL PROJECT

ELECTRICAL AND COMPUTER ENGINEERING
IOWA STATE UNIVERSITY

FINAL Project

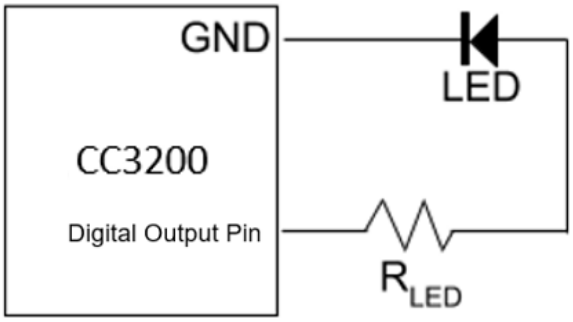
Project #1: Wireless Air Conditioning Control System.

See the grading RUBRIC below before starting with the design.

In this project, you will apply your knowledge of C to create a simple embedded system application. You will program two microcontrollers to create a "wireless air conditioning control system." The control system will function as follows: one microcontroller will act as a sensor and the other will act as the air conditioner's control system. The sensor board will collect temperature readings at a regular interval. If the temperature reading is above a certain threshold, the sensor board will tell the air conditioner to turn on over Wi-fi. The control system board is connected to a led that represents the status of the air conditioner. The led being on will represent the air conditioning being on as well. The control system board will turn on this led for a certain amount of time depending on the temperature received from the sensor.

Details:

You will write two different programs. One for the sensor and the other for the control system. Your programs will start off by establishing a connection between the two boards. This is done by connecting both boards to Wi-Fi. When both boards are connected, one board will create a server so it can receive data and the other board will connect the server as a client so it can send data to the server. It is recommended that the control system board creates the server and the sensor board is the client. Indicate That the boards are connected by printing a message to the serial monitor. Once both boards are connected, the control system can begin. The sensor will operate by receiving user input from the **serial monitor.** The user will enter the temperature into the serial monitor. If the temperature entered is above 70 degrees, then the sensor will tell the control system to turn on the led for 5 seconds. If the temperature entered is above 80 degrees, the led on the control system board will turn on for 10 seconds. The user should also be able to turn on the air conditioner manually by pressing one of the push buttons on the sensor board, you do not need to turn off the led after the button is pressed. The sensor board will send this data to the server so the control system board can receive this information. The control system board turns on and off the led using the CC3200's digital output pins. You will use 2 jumper wires to create a simple circuit of a led and a resistor. Your circuit for the control system should look something like this:



A diagram has been provided in the resources section to let you know what each pins function is. In addition, your program for the control system board will receive data that is sent to the server, interpret it, and then decide how long to turn the led on. Messages should be displayed to the serial monitor whenever an event happens to allow the user to follow what is happening.

A list of materials needed for this project are:
Two TI CC3200-LaunchpadXL's, 2 micro-usb cable, 18 board jumpers (9 per board), 2 female to female jumper wires, 2 female to male jumpers, 1 led, 1 1k ohm resistor, 1 breadboard.

The microcontrollers can be found in the lab room and should not leave the room. If you need additional materials, visit the ETG located in the first floor of cover.

Additional details and resources:

Energia IDE:
For this project you will use Energia IDE to program both microcontrollers. This is the same software used in EE 185. Energia IDE is needed to complete the project and can be downloaded by the following this link: https://energia.nu/download/

Boards Manager:
The CC3200-LaunchpadXL is not included Energia by default. Once Energia is installed, you can download the necessary packages for the CC3200 by going to the board manager. This link will tell you how to set up the CC3200: https://energia.nu/guide/boards/

Hardware Setup and Pinmaps:
To function properly the launchpads hardware must be set up properly. This link shows a picture of how the hardware needs to be setup as well as the pinmaps:
https://energia.nu/pinmaps/cc3200-launchxl/

FINAL Project

To check if your hardware is setup correctly. You can run the blink led example program under files in Energia.

This image in the hardware setup link also contains the pin-mapping of the CC3200. This tells you what is connected to each pin and will allow you to set up your pushbuttons and output pins.

Language Reference:
The Programming language was built from C and should feel very familiar to you. This link contains the syntax to language and is a reference you can use: https://energia.nu/reference/

Wi-Fi Library:
The Wi-Fi library contains all the functions you need to communicate wirelessly between the two boards. The server and client class are especially useful for this project. Each function contains a description of what it does and an example program.
https://energia.nu/guide/libraries/wifi/

**note: To do this project, you will need access to an open Wi-Fi network. This is any network you can connect to by name and password. You can use Iowa state's Wi-Fi as well, as long as you remember your password. If you have any issues stop by the ETG on the first floor of Coover. Creating a cellular hotspot to connect works for this project as well.

RUBRIC

Each of the following steps must be in its **own design file**. Demonstrate each step individually to receive credit.

a. Demonstrate an established connection between the two microcontrollers. **(20 points)**
b. Demonstrate a functional program that controls for a specified amount of time using digital output pins. **(20 points)**
c. Demonstrate a function program that allows user input through the serial monitor and pushbutton. **(20 points)**
d. Demonstrate a working program that meets all of the specified criteria. **(20 points)**
e. Final Report. **(20 points)**

If you need additional clarifications for the project, email your questions to Bsj1@iastate.edu or visit Brandon Johnson's office hours on Friday from 1:00-2:00pm in the TLA.

EE 285 FINAL PROJECT

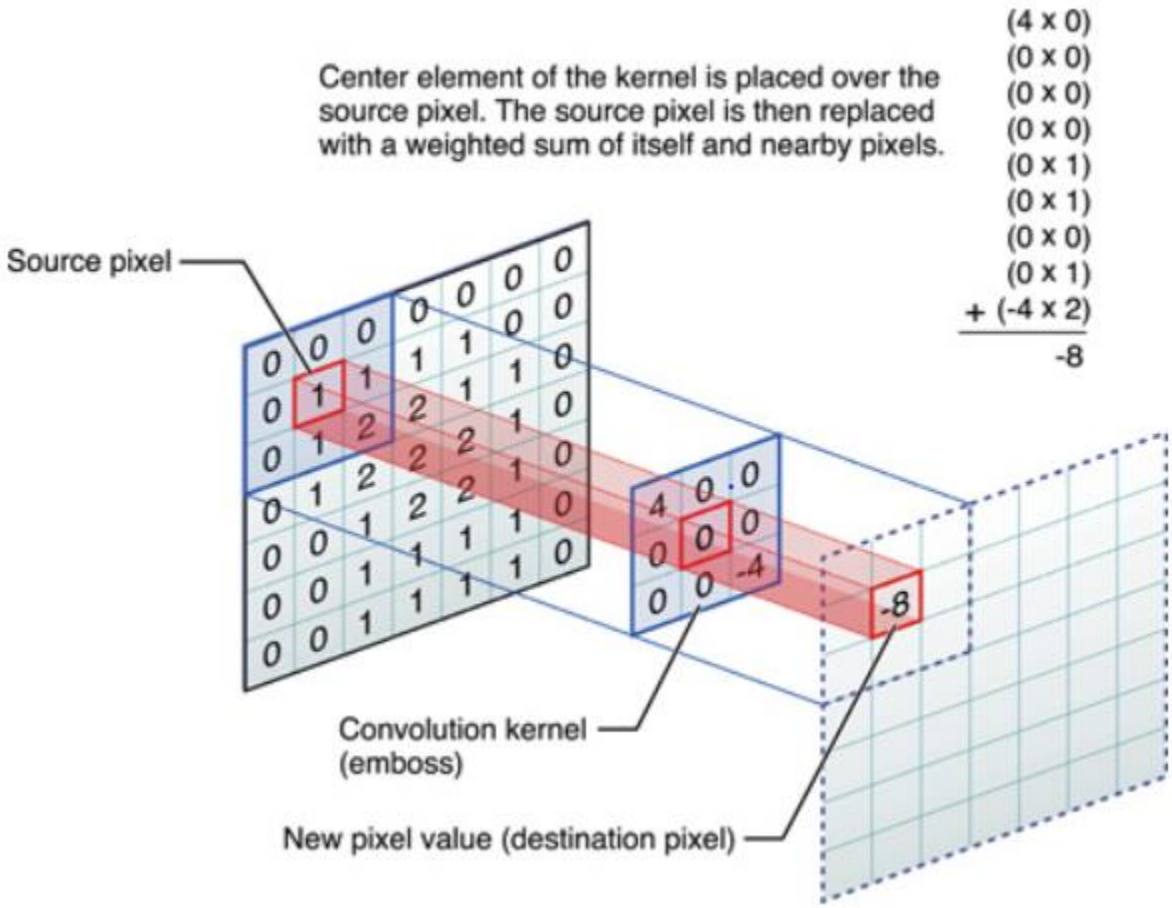
ELECTRICAL AND COMPUTER ENGINEERING
IOWA STATE UNIVERSITY

FINAL Project

Project #2: Convolutional Image Filters.

See the grading RUBRIC below before starting with the design.

In this project, you will expand the image filter program that was introduced in lab 7. Specifically, you will add several filters that involve Image convolution. Image convolution works by using a small matrix, also known as a kernel, to apply effects to images. An example of this process is shown below:



This process must be applied to every pixel in the image. Since a kernel looks at a pixel's neighbors, previously changed pixels will affect the value future pixels. You will use convolution to create three new filters: variable box blur, sharpen and edge detection. More information on these filters can be found with the following link:

FINAL Project

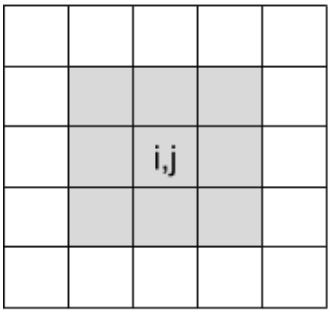
https://en.wikipedia.org/wiki/Kernel_(image_processing)

Details:

Your program will start off by displaying a menu shows each filter and asks the user what filter they would like to apply. The user should be able to apply multiple filters before outputting the new image. Specifications for the variable box blur, sharpen and edge detection filter are listed below:

Variable box blur:

You can blur an image by setting each pixel to the average input values of that pixel and the surrounding pixels. For example, in the below pixel array, the blurred RGB values for pixel (i,j) would be the average RGB values of all of the shaded pixels.



The blur filter described above only averages the given pixel and the surrounding 8 pixels. You might say that the blur has a “radius” of 1, because all pixels within one row and/or column of the given pixel are included in the average. We can increase the magnitude of our blur filter by increasing the “radius” of our blur.

Add a control to your blur filter that allows you to choose the “radius” of the blur. For example, if the user chooses a “radius” of 5, then all pixels within 5 rows and/or columns of the given pixel (yielding a square of size 11x11 pixels) should be included in the average for that pixel. This “radius” value should be a parameter to the blur filter function.

Sharpen:

While blur filter makes each pixel value more similar its neighbors, a sharpen filter essentially does the opposite by making each pixel more unique. You will need to use a kernel that applies this effect. Cy.BMP is not a good image to show the functionality of this filter. Use a different image with less resolution when demonstrating this filter.

EE 285 FINAL PROJECT
ELECTRICAL AND COMPUTER ENGINEERING
IOWA STATE UNIVERSITY

FINAL Project

Edge detection:

And edge detection filter locates all the edges in an image. Your filter should output a black and white image where all defined edges are white and everything else is black. The way you will do this is by applying the Sobel operator. Additional information about the Sobel operator can be found with the following link: https://en.wikipedia.org/wiki/Sobel_operator . The Sobel operator will not create a black and white image by itself, you will need to create a threshold to determine if each pixel can be classified as an edge or not.

In addition to meeting the requirements above, your filters need to handle edge cases, pixels where the kernel is out of the image's bounds. An example of edge cases are the corner pixels and pixels in the first and last row and column. Also, some of your filters might produce values that are larger than the max RGB value of 255. A way of normalizing the RGB values so that they are between 0 and 255 should be implemented. Finally, all filters need to be in the img header/source file that was created in lab 8:

RUBRIC

Demonstrate each step individually to receive credit.

a. Demonstrate that all three filters work as specified. **(45 points)**
b. Demonstrate that your program handles edge cases. **(15 points)**
c. All functions in a separate header/source file and demonstrate that multiple filters can be applied. **(20 points)**
d. Final Report. **(20 points)**

EE 285 FINAL PROJECT
ELECTRICAL AND COMPUTER ENGINEERING
IOWA STATE UNIVERSITY

FINAL Project

Project #3: Specify Your Own Project

See the grading RUBRIC below before starting with the design.

Propose a project of your own design. Once again, the minimum constraints are that your project MUST: 1) include pointers 2) include functions, and 3) a custom library, a separate source/header file.

If you choose this option, you **MUST** discuss your intended design with your lab TA to see if the project would be appropriate for this class. This discussion can be over e-mail. Please contact your lab TA.

If the lab TA or Instructor evaluates the project and recommends that it is not of appropriate difficulty, then the project will be denied and you'll have to modify it or propose a new one.

RUBRIC
Demonstrate each step individually to receive credit.

a. Design and demonstrate the functions for your project **(20 points)**

b. Demonstrate how your program uses pointers **(20 points)**

c. Demonstrate a well-organized program that uses a custom library **(20 points)**

d. Put together all of the individual components in Parts a,b, and c and demonstrate the complete project. **(20 points)**

e. Final Report. **(20 points)**

EE 285 FINAL PROJECT

ELECTRICAL AND COMPUTER ENGINEERING

IOWA STATE UNIVERSITY

FINAL Project