Julian J. Peters
petersj2@wit.edu | (239)-810-4115 | Boston, MA
Available for Fall 2019

Project Portfolio

Personal Projects

WS2812 LED Library for ATTINY85 in C

- Utilized ATTINY85 and WS2812 datasheets to understand how the two would communicate
- Wrote C as well as AVR assembly code to accurately control GPIO pins
- Created LED animations with the final library

PCB for 16-key shortcut board

- Designed a PCB that would utilize an ATmega32u4 microcontroller to send HID inputs to a computer using USB
- Used schematic entry in Circuit Maker to create the design containing the keys, micro-USB port, resistors, diodes and capacitors to allow the ATmega32u4 to accept USB power, input from the keys, and a clock from the on board crystal.
- Programmed the ATmega32u4 using the on board SPI interface and loaded a program onto an external Arduino that would act as the master for programmming the board.
- Used the Arduino IDE to write C/C++ code that would send HID commands to a connected PC. The code allowed for the user to press a key and possibly type in their entire password for websites or automatically open an application and generally improve productivity.

ESP8266 & Google Assistant Light Control

- Created multiple voltage levels from 12 volt source to power both the LED strip as well as the ESP8266 board using a buck converter
- Integrated logic level MOSFET to control LED strip power and light level with PWM
- Utilized knowledge of C and MQTT to make the ESP8266 communicate with the cloudmqtt services
- Final product was able to take commands from the google assitant and pass them to the ESP8266 to control the LED strip power as well as light level

File Organizer for Windows

- Used C to read in files within a download folder and categorize them by their extension
- In order to work with the windows file names and take into account the varying size of the downloads folder pointer manipulation and dynamic memory allocation was used

Google Home MQTT Computer Control

- With the use of Python, IFTT and a Node.js server a user could speak to their Google Home to open a specific webpage, search Google, search Youtube or open any other application on their computer.
- Wrote Python code that will subscribe to an MQTT broker and perform various tasks based on the state that the broker reported back to the client PC.
- Created a Node.js server using Javascript that acted as a MQTT broker that would receive JSON data from IFTT. Using the JSON data, the broker could update the subscribed clients and Python would take care of what the PC should do.
- The Google Home already has integration with IFTT so a few simple applets needed to be created containing the proper JSON data depending on speech received from the Google Home

Python Webscraper

- Used Python and Reddit.com's integrated API I was able to build an app that would download the wallpapers from Reddit.com/r/wallpapers.
- By parsing all of the information that Reddit's API returns to Python I was able to make specific seletions from the webpage and only download images that had resolutions matching my particular monitor.

Python Pool Helper

- Used Python and OpenCV I was able to capture images from a pool game and return back an image that highlighted the path that the ball would take to assist with shots taken
- Implemented OpenCV's canny edge detection and other image manipulation tools to highlight the path the ball would take as it is shot

Node.js Twitter Bot

- Used a Node.js server and packages to communicate with a Sonos speaker to automatically tweet out the songs that were playing as well as album artwork

Personal Projects Continued

Machine Learning Online

- Used online resources like Lynda.com and Udemy.com to learn machine learning principles
- Learned to use Jupyter notebooks, Pandas, Keras, and Tensorflow to create machine learning algorithms
- Learned about different activation algorithms like ReLu and sigmoid
- Learned about Regression models, Tree models and Convolutional Neural Networks

Other Academic Projects

Verilog FPGA Name Display | Advanced Digital Circuit Design

- Wrote Verilog code that would enable control of the 7-segment displays built in to the development board. Using the switches as a 3-bit binary input a name could be displayed letter by letter as the number increased.
- Equations were written in a dataflow format with continuous assignments to the pins and Quartus Prime was used as an IDE to write, debug, and upload the code as well as assign inputs and outputs to pins of the FPGA.

ARM Processor Emulation in Python | Computer Architecture

- Utilized knowledge of the ARM processor architecture to write a program that will accept a text file of instructions and execute them and fill virtual data memory and register memory with the appropriate values.
- Implemented all of the basic instruction types as well as conditional and unconditional branching. Also added features like pipeline registers to support pipelining and forwarding to speedup the processor.

Multi-player Battleship Game | Object Oriented Programming

- Implemented knowledge of objects and classes to form a feature-full game of battleship using C++
- Worked collaboratively to make a game that showcases inheritance, function overloading and other Object Oriented fundamentals

Side-scrolling Game in C | Microcontrollers

- Utilized multiple functions and external files to create a character texture as well as moving obstacle textures
- Implemented interrupt service routine to allow for player input and pointers to manipulate timing registers for updating the screen
- Worked collaboratively and individually to troubleshoot problems and come up with unique methods to draw what we envisioned