Justin T. Conroy

801 S. Plymouth Ct. #216
Chicago, IL 60605
(708) 469-9246
Justin@Conroy.in

WORK HISTORY

Firmware Engineer Contractor July 2023-Present
Ronin Surgical Remote/Chicago, IL

At Ronin, I developed firmware to control their surgical headlamp product. The product was built on an ESP32. It was written in C++ and used FreeRTOS as a base. The firmware used an SMBus connection to do smart battery management. It also incorporated feedback from the high powered light using a custom Protocol to manage fault conditions.

Firmware Engineer Contractor October 2020-November 2022
Kinoo Remote/Chicago, IL

At Kinoo, I developed firmware for the Kinoo Wand. An educational toy and game controller powered by an ESP32. This project was written in C++, and used FreeRTOS as a base. It included WiFi and Bluetooth Low Energy (BLE) connectivity. It also included some multicolored LEDs which played patterns synced to music which was played on an onboard speaker. I designed a custom file format with FlatBuffers to allow designers to create complex patterns and upload them to the device filesystem (LittleFS). The controller also included an Inertial Measurement Unit (IMU), which included a gyroscope, accelerometer, and magnetometer. The firmware processed data from this hardware to do complex gesture recognition. In addition, the firmware was able to update itself using Over-The-Air (OTA) updates. I also wrote custom code for managing updates to media files on the device over the internet.

Principal Software Engineer September 2016-August 2019
Koya Law Chicago, IL

At Koya, I was the lead backend developer for a custom web application used by the company and their clients. I designed and implemented a custom REST API to be consumed by a javascript frontend created by the Frontend Developer. The backend was written primarily in C# and ASP.NET Core, with MS SQL Server as a supporting database. The application ran on an Azure instance.

Application Developer September 2013-September 2016
Valence Health Chicago, IL

At Valence, I mostly did backend development ETL (Extract, Transform, Load) tasks. The company deals in healthcare insurance and I was generally responsible for a lot of data coming in and going out to vendors in a multitude of different formats. I used C# and Entity Framework to marshall data between files, web services, and databases. I also worked on some user-facing applications in ASP.NET to create tools for operations support staff to work more efficiently. In this line of work, I dealt with a lot of Patient Health Information and I take privacy of that information very seriously.

Software Engineer September 2011-August 2013
Elettric80, Inc. Chicago, IL

At Elettric80, I developed software for controlling laser guided vehicle systems, conveyor systems, and other automation machinery. I also developed, installed, and maintained warehouse management systems which integrated with the robotic platforms Elettric80 develops. The software I developed was written in C# and SQL Server and communicated with robots, PLC systems, and customer ERP systems. I worked directly with high profile customers to ensure the quality of the product we delivered and to implement custom features for every installation.

Engineering Intern June-September 2011
Marki Microwave Morgan Hill, CA

I designed and implemented embedded software for a new product that Marki Microwave released. The product was a piece of lab test equipment which interfaced wirelessly with software on a PC to provide test data to users. I wrote an interface to LabView as well and provided a full API for interacting with the device to allow users to write their own programs for the device.

EDUCATION

Bachelor of Science, Computer Engineering
University of Illinois, Urbana, IL
graduated December 2010

COMPUTER SKILLS

Languages: C++, C, C#, Python, LaTeX
Protocols: I^2C, I^2S, SPI, FlatBuffers, MQTT, SMBus
Platforms: ESP32, FreeRTOS, ASP.NET Core, Entity Framework, LINQ, SQL Server
Version Control: git, SVN
Compilers: Visual Studio, GCC
Operating Systems: Windows, Linux, OSX
File Systems: LittleFS, SPIFFS
Other Technologies: ADC, DAC, Bluetooth Low Energy (BLE), WiFi