

# Johnathan J. Flaggs

LinkedIn: [www.linkedin.com/in/johnathanflaggs](http://www.linkedin.com/in/johnathanflaggs)

Website: [www.johnathanflaggs.com](http://www.johnathanflaggs.com)

Email: [johnathanflaggs@gmail.com](mailto:johnathanflaggs@gmail.com)

Phone: [+1.949.414.9545](tel:+19494149545)

---

Software Architect with a focus on backend system development for fintech and robotics applications. Successful record of leading projects from proof of concept into production. Professional references available upon request.

## Education

### University of California, Riverside & Davis

Sep. 2010-Jun. 2014

- UCR: BSME with concentration in Control Theory under Department of Mechanical Engineering (BCOE)
- Control Theory Courses under Department of Mechanical and Aerospace Engineering

## Technical Toolset

### Development Languages & Environments:

- |                       |                    |               |                  |                 |
|-----------------------|--------------------|---------------|------------------|-----------------|
| ▪ Visual Studio       | ▪ Codeblocks       | ▪ SVN/GIT     | ▪ Batch Script   | ▪ Node.js       |
| ▪ Atmel Studio        | ▪ C, C++           | ▪ C# .NET     | ▪ MATLAB         | ▪ HTML, CSS, JS |
| ▪ RSLogix/FactoryTalk | ▪ Beckhoff TwinCAT | ▪ MagneMotion | ▪ Fanuc Robotics | ▪ Cognex Vision |

## Professional Experience

### Lead Software Architect at Essentium 3D

Jun. 2021-Present

Leading robotics software architecture and development for high-speed industrial 3D printers. Ensuring that our team builds a software core that is sufficiently robust, scalable, and maintainable to support the product line.

- **Lead** the design and architecture of our core software which allowed us to expand from plastic to metal printing
- **Mentoring** Jr. Engineers on approaching complex problems and upkeeping best coding standards
- **OOP** and heavy emphasis on robustness, scalability, maintainability, and patterns in C# and C++
- **Anticipate** and mitigate the impact of future design changes on the software layer
- **Design** of macro and micro software architectures which define the core product
- **Manage** and review source control on a per-commit basis

### Engineering Lead at Harpoon Protocol

Jun. 2020-Jun.2021

Leading architecture and development for a loan liquidations (DeFi) platform. The platform allows users to manually participate in the liquidations market and hosts a proprietary Smart Liquidation Algorithm to provide liquidity to the market. I chose to use Node.js, C#, and React as our primary technology stack.

- **Designed** and trained a predictive model to compete in a time-only based order book
- **Consulted** for Anchor Protocol (Terraform Labs) on improving their liquidation order book
- **Contributed** executive decisions on software features that influence product direction
- **Managed** and supported a team of Sr. and Jr. level developers on approaching complex blockchain concepts
- **Anticipated** and mitigated the impact of API changes on the software layer
- **Educated** investors and stakeholders on the current state and future development of our product

## Professional Experience

### Lead Software Engineer at Satsy

Mar. 2019-Jun. 2021

Developed an algorithmic trading platform which integrates multiple brokerage API's for live, simulated, and historical trading/analysis. The platform is multi-threaded and supports many unique features that are not available via popular trading platforms.

- **Developed** an easy-to-use strategy/indicator language that is unlimited in extensibility
- **Multi-Threaded C#** application allows real time data streams and an intuitive systems-focused UI
- **Created** a thread-safe real-time library for advanced mathematics using time series data
- **Low-Latency** execution C# library benchmarked against C/C++ for performance testing
- **Modular** architecture allows any UI to consume the Satsy.App backend service
- **Supports** live and simulated trading with TD Ameritrade, Binance, Gain Capital, and Simulated data feeds
- **Developed** a custom charting solution for rendering up to 250k data points in view, and > 2M points in memory

### Sr. Robotics Software Engineer at Seagate Technology

Mar. 2017-Mar. 2019

Developing software for cutting edge processes in digital storage technology. My contributions include:

- **Windows C# .NET** proprietary machine and vision process controls
- **Support** core vision libraries in C/C++
- **Cognex** VisionPro API integration
- **Machine-Vision Calibration** to establish precision robot coordinates
- **Motion Control** Kinematics, pick-n-place, multi-axis coordination, Quantum HSM framework
- **User Interface** Allowing users to fluidly interact with the multi-threaded application
- **Version Control** Using SVN, TFS and Agile/Scrum using Jira
- **OOP** Heavy emphasis on encapsulation, inheritance, polymorphism, and robust design patterns

### Retail Trade Systems Developer

Jan. 2016-Mar. 2018

Developed discretionary and semi-automated trading systems as a self-funded trader. It was during this time period that I developed a systematic approach to analyzing and trading in the financial markets.

- **Developed** indicators and strategies using Tradestation EasyLanguage
- **Built** a framework built on NinjaTrader 8 to feed my statistical models hosted in an external C# .NET application
- **Scripted** analysis tools for statistical research in MATLAB, C++, and .NET to find an edge in price data

### Lead Controls Engineer at Sorenson Engineering Inc.

Sep. 2014-Mar. 2016

Leading controls software development for high-volume manufacturing. I developed core software and worked closely with Mechanical Engineers to build proof of concept products (R&D environment).

- **Fieldbus Integration** – Integrating third-party hardware/software nodes into a controls network
- **Motion Control** – Kinematics, motor sizing, pick-n-place, multi-axis synchronization, and axis coupling
- **Vision Inspection** – Driving digital cameras (Cognex, Baumer/VeriSens) via native C++ SDKs
- **Eliminated** need for expensive camming software license and a measurement sensor.
- **Increased** machine PPM by decreasing rotor inertia ratio by 140% for tighter position control (VSIII)