Callum Dyson-Gainsborough

.NET Software Engineer

Lincolnshire, UK | **GitHub:** github.com/CallyyllaC | **LinkedIn:** linkedin.com/in/callum-d-03168515b/ | **Web:** awoo.ltd
Contact via LinkedIn. Direct contact details available on request.

PROFILE

I'm a .NET Software Engineer with nearly six years' experience working mainly in the security sector. Most of my work is on long-lived .NET Framework and modern .NET applications where the software has to work with real hardware, not just other software. I work across frontend, backend, image processing and hardware integration, and I am usually given a feature or problem to take from the initial requirement through implementation, testing and support.

TECHNICAL SKILLS

Languages & Platforms	C#, C++, Python; .NET Framework 4.x, .NET 10
Desktop & Architecture	WPF, Avalonia UI, Windows Forms, WCF, Windows Workflow Foundation, MVVM, Castle Windsor, async producer/consumer architectures
Performance	Concurrent and parallel algorithm design, SIMD/vectorisation, high-throughput acquisition, near-real-time processing, dotTrace profiling, memory and I/O optimisation
Low-level & Hardware	Unsafe C#, P/Invoke/native interop, hardware abstraction layers, serial/network communications, device APIs, calibration and full software-to-hardware debugging
Imaging	Image processing and manipulation, edge detection, bad-pixel detection, calibration/correction algorithms and scale-aware image analysis
Web, Data & Testing	HTML, CSS, JavaScript, SQL Server; NUnit, NMock, FluentAssertions, AwesomeAssertions
Tools	Git, GitHub, SVN, Visual Studio, Rider, TeamCity, JIRA, TestRail, Confluence

PROFESSIONAL EXPERIENCE

Software Engineer | R&D | Security Sector

Nov 2020 - Present

- Main software engineer responsible for bringing amorphous-silicon 2D detectors into a mature codebase that had previously been built around 1D line-scan systems. This meant changing the acquisition, calibration, processing, WCF and frontend workflows rather than simply adding support for a higher-resolution detector.
- Integrated three detector models from two suppliers. I wrote the majority of the backend used for acquisition, calibration and processing, and heavily modified the shared middleware and frontend so the same product could support both line-scan and 2D hardware.
- I am the main internal contact for the 2D detector platform and support R&D, production, sales and technical support when something needs implementing, commissioning or debugging.
- Wrote in-house calibration and correction methods where generic approaches were not suitable for the underlying image data, including bad-pixel detection and correction that adapts to acquisition conditions, electronic noise and device temperature.
- The move from streamed line-scan data to 4K+ 2D images exposed performance problems that previously did not matter. I worked on processing speed, memory use and leaks, calibration time, caching, and file loading/saving using profiling, parallel processing, async pipelines, SIMD and lower-level optimisation where it was justified.
- For an automated detection requirement, tested several classical image-analysis approaches before building a chamfer edge-matching system from raw image through to frontend overlays and template management. I also wrote automated tuning tools and later integrated a third-party neural-network engine behind the same frontend workflow.
- Port mathematical and image-processing code from MATLAB into production C#, and have implemented runtime DLL loading, native interop, hardware abstraction and unattended acquisition paths where new work has had to fit around an established legacy architecture.
- Worked extensively on the Python/Linux side of new pulsed-generator integration and regularly debug problems across the whole product pipeline, including application code, networking, operating systems, device firmware and physical hardware.
- Maintain shared TeamCity builds and internal engineering tools. This has included moving build infrastructure to produce both .NET Framework 4.8 and .NET 10 packages, commissioning/calibration tools, licensing workflow improvements and JIRA automation for recurring engineering and operational tasks.
- Provide technical support to international partners, including remote work across several regions and on-site support in Poland during final customer acceptance and handover preparation. I also write commissioning/testing documentation, contribute to user manuals, deliver internal and partner training, and take part in FAT/TestRail testing.

Additional responsibility: Health & Safety Representative and Fire Marshal, covering staff briefings and refreshers, safety communication, and routine checks of workplace fire-safety provisions.

SELECTED PROJECTS

Noctaxis

.NET 10 | C# | Avalonia

Started Noctaxis after finding a gap for an astrophotography planner that was capable without being tied to an expensive or intrusive subscription model. The aim is a completely free desktop application built from freely licensable libraries and data sources. The main technical work so far includes camera/lens field-of-view planning, terrain-aware visibility and cone calculations, terrain-data integration, and turning the results into a polished planning interface rather than a collection of technical tools.

The Ember Deck

Python | Raspberry Pi | Linux

Built as a practical way to teach myself more about electronics and hardware I/O. The project turns a dead vintage TV/radio/cassette unit into a Raspberry Pi media console, combining Python with GPIO/I2C peripherals, analogue controls, displays, audio hardware, LEDs, networking and custom power distribution. The software is split into supervised worker processes so hardware access, media control, shared state and audio visualisation can fail and recover independently.
github.com/CallyyllaC/The-Ember-Deck

Leap Motion Robot Telemanipulation

C++ | Python | ROS | MoveIt | Linux

Final-year university project investigating whether multiple Leap Motion controllers were practical for robot telemanipulation. Built a ROS prototype that mapped hand position and gestures to a simulated Panda arm using custom C++/Python nodes, messaging, coordinate conversion, gripper control and a movable/scalable virtual workspace. Because the available Linux drivers were old and could not properly support multiple devices on one machine, I worked around the limitation by running controllers on separate networked systems and combining their ROS data. The project worked, but the final evaluation concluded that the tracking noise and driver reliability made the hardware unsuitable for serious use.
github.com/CallyyllaC/LeapMotion-Telemanipulation

EDUCATION

University of Lincoln - BSc (Hons) Computer Science, First Class

2017 - 2020

Selected modules included Parallel Programming, Autonomous Mobile Robotics, Machine Learning, Artificial Intelligence, Software Engineering, Networks and Network Systems, Database Systems and Object-Oriented Programming.

North Lindsey College - BTEC Level 3 Extended Diploma in Computing, DDM

2015 - 2017