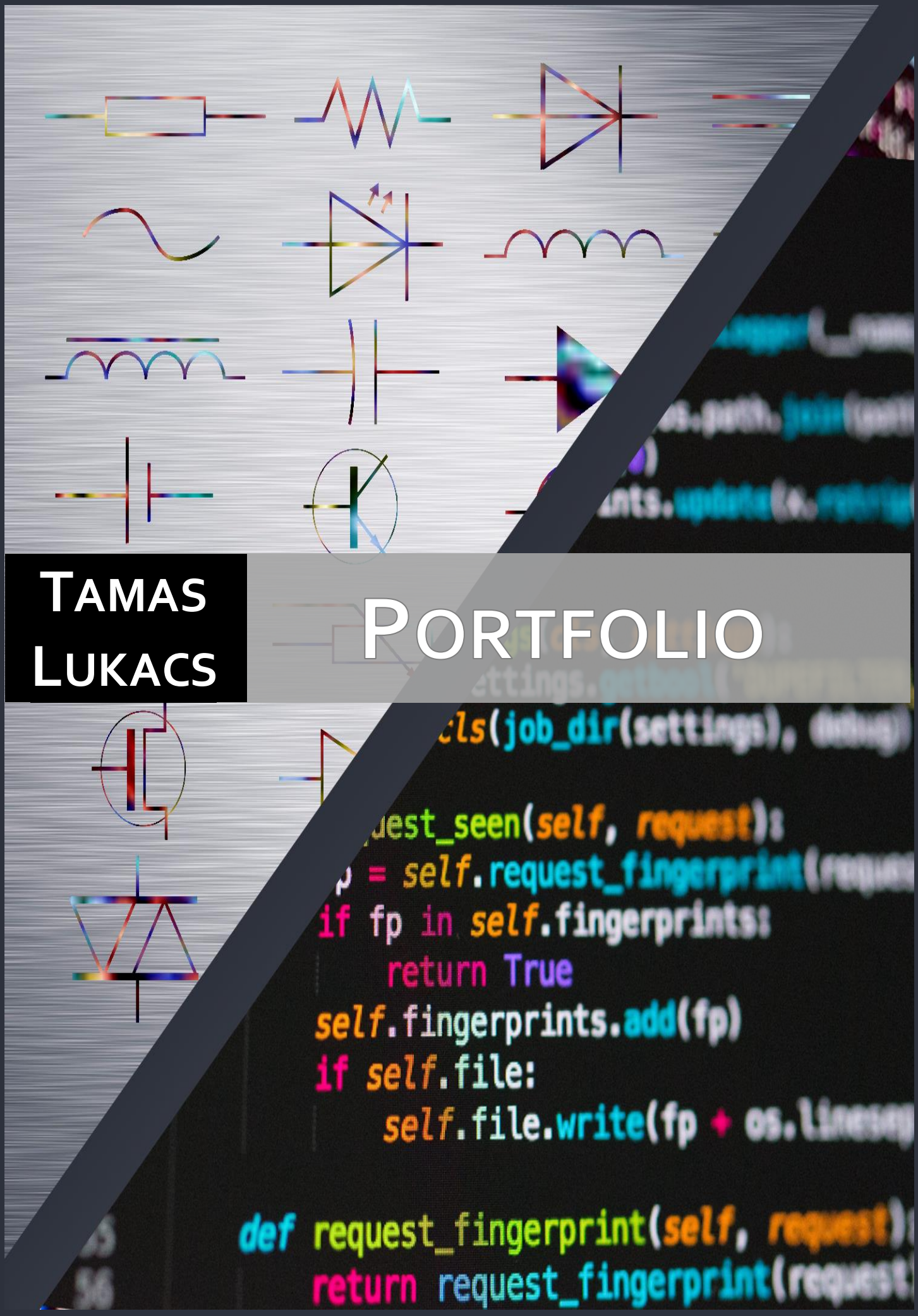
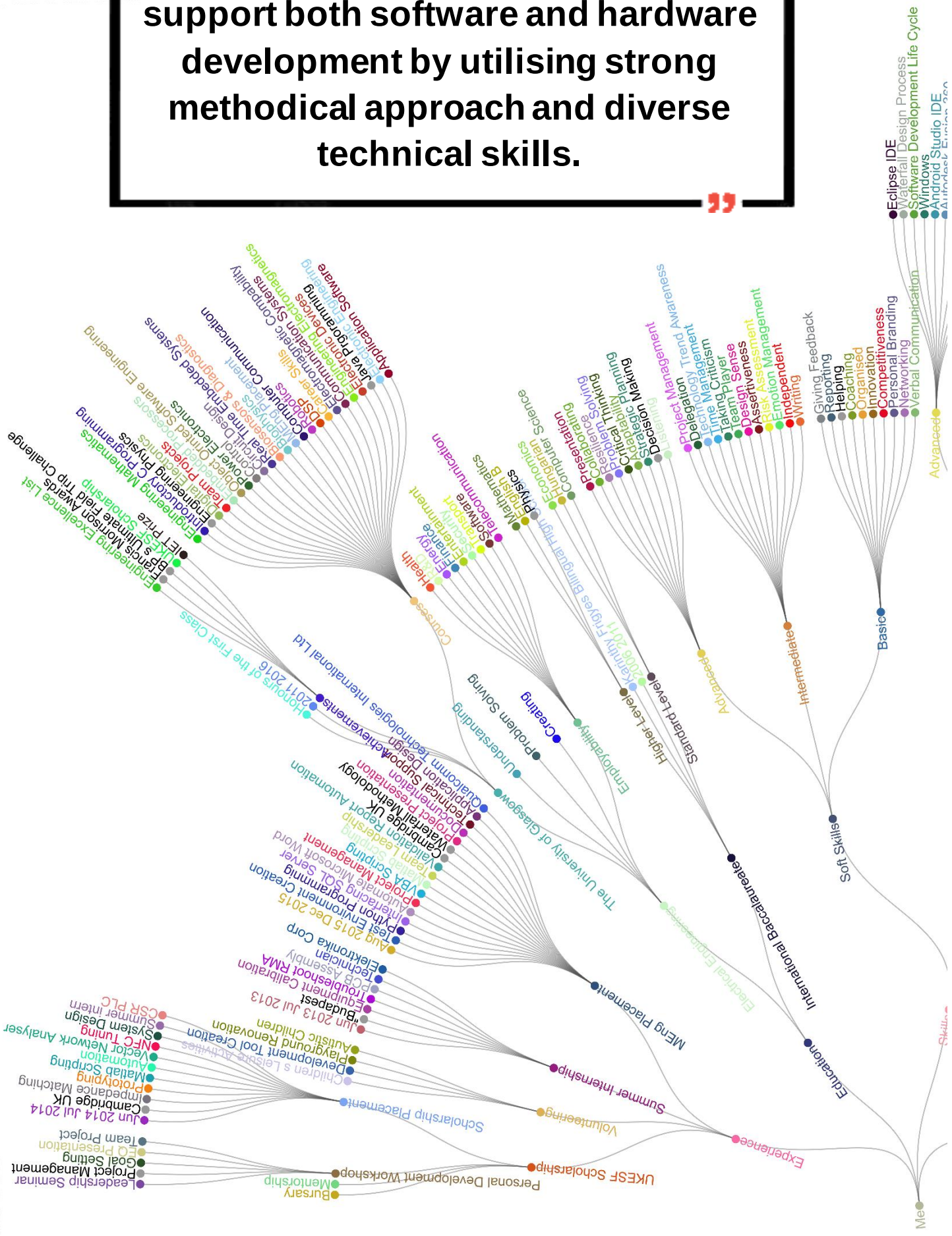
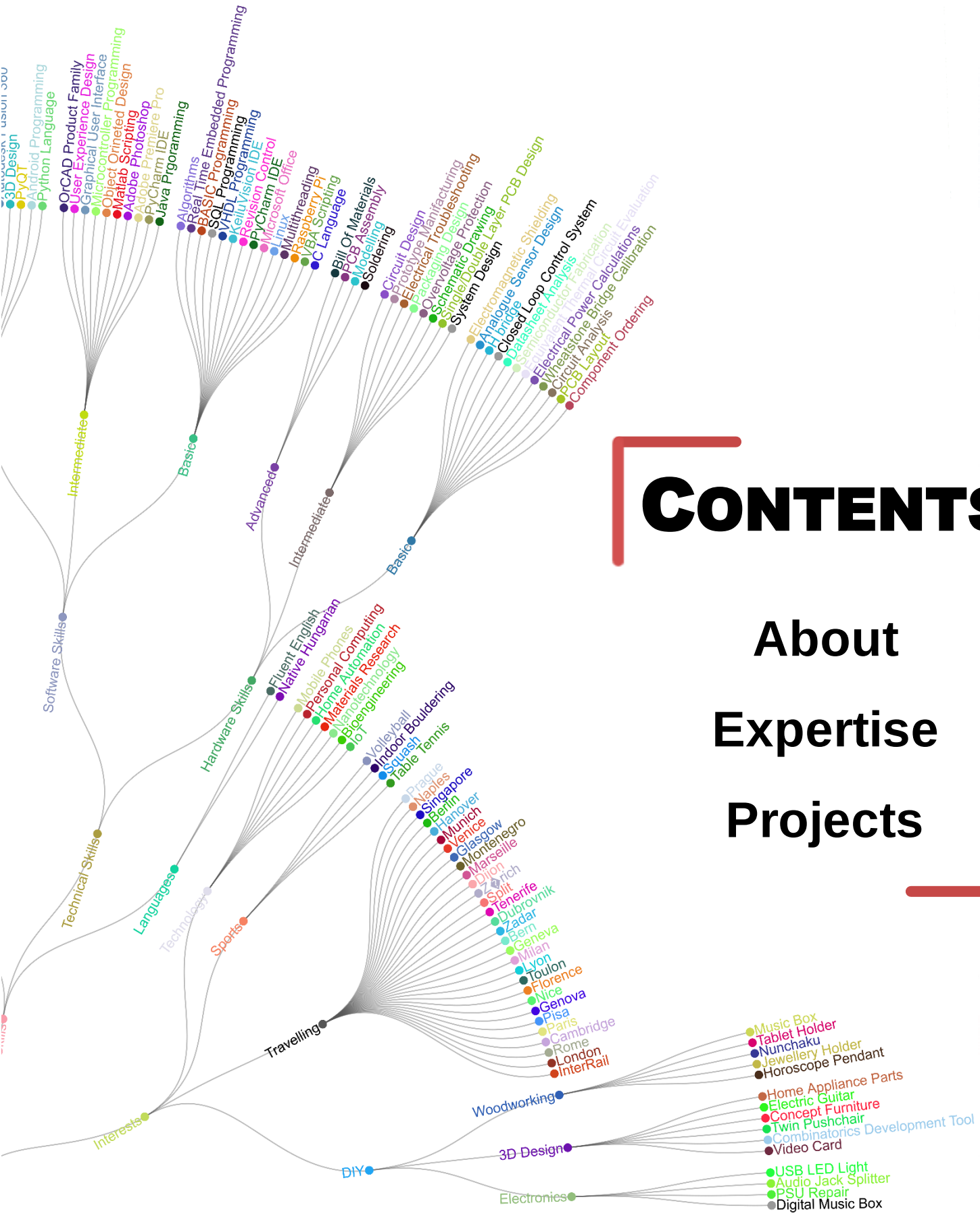

Tamas
Lukacs
Portfolio
def request_fingerprint(self, request):
return request_fingerprint(request)
if fp in self.fingerprints:
return True
self.fingerprints.add(fp)
if self.file:
self.file.write(fp

Senior Engineer thriving on innovation with eagerness to support both software and hardware development by utilising strong methodical approach and diverse technical skills.

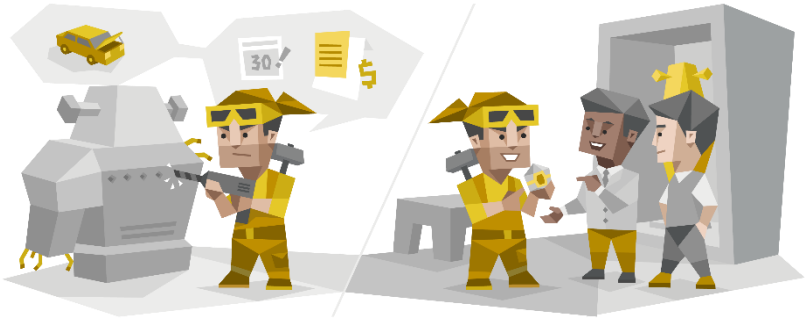


lukacstomi91@gmail.com Resume /tamas-lukacs



Contents

About

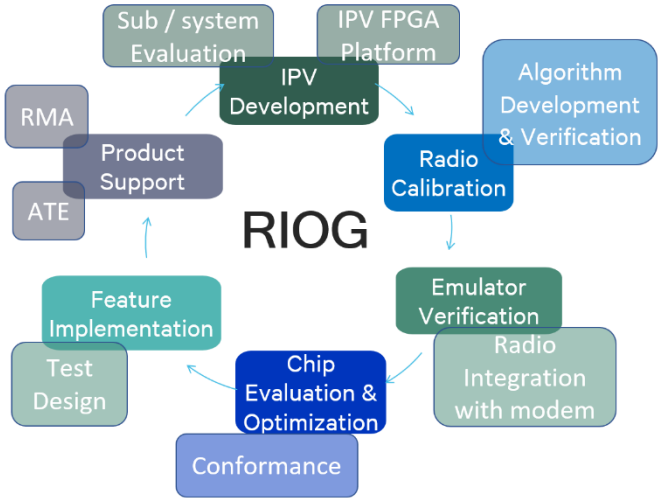


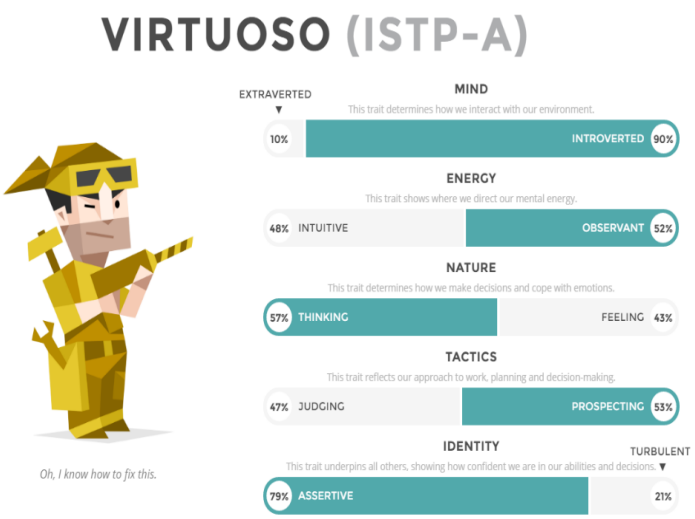
Hi, I'm [Tɒmaːʃ]!

I am a UK based Senior Engineer with 5 years of industry experience at Qualcomm. After finishing my studies at University of Glasgow with honours of the first class in Meng Electronics and Electrical Engineering, I have started my career at Radio Integration and Optimisation Group (*RIOG*) in Cambridge. As a member of the group, I have been exposed to a highly dynamic environment revolving around product development of radio equipped silicon solutions. Throughout the years, I have played a vital role in the evaluation, verification and testing of such products while also being successful in further expanding on the group's responsibilities, which now touches on every aspect of product development of Qualcomm's Bluetooth chipsets.

Having been through multiple chip projects, I have gathered tremendous experience from each stage of the production; my favourite being feature implementation where I constantly have to battle with the unknown and new. Regardless of the role, my vast experience in automation and programming proved to be always critical in excelling at the task at hand.

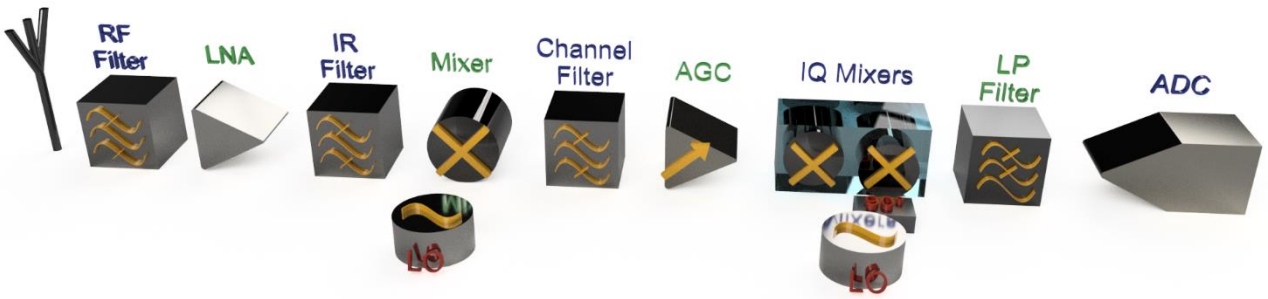
Moreover, working in one of the largest groups of 19 members on board, as well as getting involved with cross-group and cross-site projects, not only ensured the cultivation of a wide range of skillset I can deploy, but it also solidified my communication and people skills while enabling me to exercise my leadership muscles.





I am genuinely interested in programming even outside of work and focused on keeping myself updated with the latest trends regarding the new practices in object-oriented approaches and how Python is used in the field of data science.

The challenges I face in this field are definitely the right fit for my personality type of ISTP (Introverted, Sensing, Thinking, Perceiving), and my passion for creating novel solutions fundamentally aligns with any prospective team in the engineering sector who are looking to innovate and set a high level of standard.



Radio Integration and Beyond

Team and Individual Productivity

Since radio up brining is no small feat, I have naturally taken part in multiple projects where team effort and collaboration are critical, as well as taking on individual assignments related to feature debugging or tooling, where I have been left at my own devices. I tackle both scenarios equally well by providing candid feedback while being concise and assertive in a group environment and showing transparency while enabling my creativity to innovate. I also enjoy mentoring which helps to empower and distribute knowledge among the team.

Software Engineering

Evaluation and debugging as well as creating the tools to facilitate these activities were enabled by my extensive Python knowledge which branches out to areas including automation of test equipment, data handling, GUI implementation, report generation and more. I am also familiar with C, Android, Java, MATLAB, SQL, VBA, BASIC, LaTeX and HTML which grants me a broader perspective. This, combined with industry experience of continuous integration with Git and Perforce enable me to cultivate good coding practices.

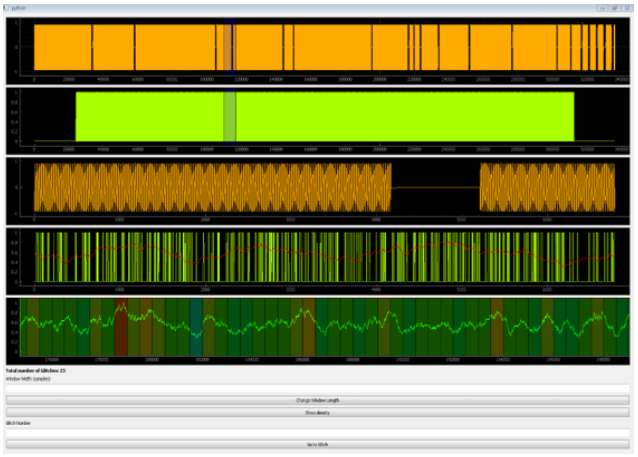
Technical Knowledge

My solid electronics background and my ability to quickly and effectively grasp new concepts has helped me to tackle multiple challenges related to individual building blocks of the radio such as varying DC offsets and the calibration of it at the input of ADC or degraded SNR performance due to clock synchronisation issue. I have also evaluated RF system and protocol level performance including sensitivity or adaptive frequency hopping. Moreover, I have a well-established understanding on several digital communications such as I^2C, SPI or CAN.

Methodical Approach

My thorough analysis, good sense of prioritisation and systematic exploration towards the indefinite are my strong suit and proved to be instrumental in keeping to tight deadlines and conquering the unknown. Having experienced both agile and waterfall methodologies and being scrum master certified, I am confident that I can deliver.

Projects



Key Highlights -- RIOG

Interference Robustness

Impact ★★★★★

Difficulty ★★★★☆

[Python, C++, pyqtgraph, numpy, scipy, multiprocessing]

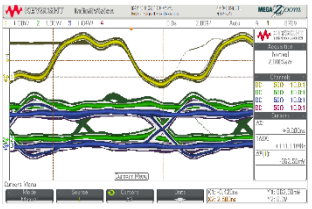
One of two engineers who designed Python driven test framework from scratch and actively participated in customer discussions related to reliable audio testing of Bluetooth products in the presence of interference. First one to quantifiably describe user experience and relate it to RF and firmware behaviour. Effort was focused on modularity, high-level automation, research, programming in Python and embedded C and post-processing. Key challenges included large quantities of audio manipulation and cross correlation of events. Major benefits included unblocking product release, high-end customer satisfaction and engagement on future solutions as well as gaining the ability to benchmark manufacturers' designs and preventing firmware bugs to reach customers.

RX Test Lead

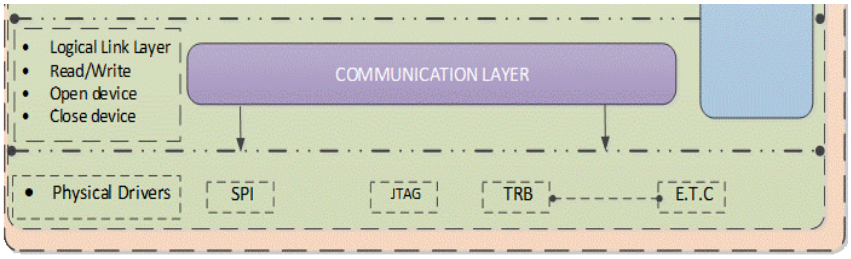
Impact ★★★★☆

Difficulty ★★★★★

[Python, pandas, PIL, pyvisa, pptx, matplotlib]



First IoT opportunity at Cambridge site whereby RIOG took was fully involved even in the emulation phase to mitigate risks. Responsibilities included orchestrating the collection and improvement of receiver's key performance measures, gathering and ensuring BT-PHY test specification compliance, test design, programming and regression integration, report generation, mitigating remote setups by utilising on-site technicians and working together with designers on debugging performance issues. One of the major challenges I undertook was sensitivity degradation from emulator platform to ASIC which has been successfully tacked by extensive cooperation with the design team. This was also the first project that implemented from start to finish my solution where test data is recoded in a company wide database tool. It had reduced the time to generate reports by at least a third and introduced commonality between each test lead, ultimately contributing to the chip being successfully delivered in the shortest timeframe of eight weeks versus the traditional 16 weeks, accelerating contract win.



RIOG -- Key Highlights

Common Scripting Executor

Impact

Difficulty

[Python, pypandoc, pylink, Unittest, SPI, dunder methods]

One of four-member team on a mission across sites and groups to unify environment and methodologies of interfacing with chipsets as an effort to cut down time until chip sampling. I took full ownership over communication layer of the Python tool where responsibilities included all stages of the software development life cycle including writing up software requirements specification and developer notes, architecture and API design, implement and test a variety of communication protocols such as SPI and SWD, and continuously maintain and expand original design as requirements evolved. I am also responsible for a neat implementation to enable multi-connection support requested as the spec changed. Traditional implementation would have meant redesign of each component layer, but by using magic functions and implementing a top shim layer, no reengineering was needed. After successful proof of concept, CSE is targeted to phase out existing solutions for the groups involved for the next generation of chipsets in late 2021.

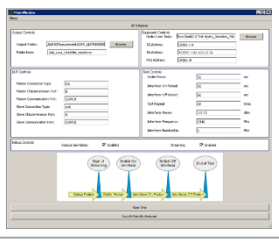
Adaptive Frequency Hopping

Impact

Difficulty

[Python, subprocess, PIL, re, matplotlib, numpy, PyQt4]

As my first individual project, I have taken on the evaluation of one of the core techniques for a transceiver to cope with interference, Adaptive Frequency Hopping (AFH). Customer driven issues paved the way for new responsibility within group to research, analyse and debug the performance of the mechanism. As a sole owner of the task, responsibilities meant getting familiar with the Bluetooth Special Interest Group standard, understanding existing implementation and fabricating means to debug behaviour for multitude of parameters. I have developed a novel way to represent exhibited behaviour though GUI tool which helped not only to identify and fix firmware bugs but also further enhance the algorithm. The work was used as a base for additional improvements which now boost performance across all chipsets of Qualcomm.



Key Highlights -- Academic Work

Smart Bike Concept

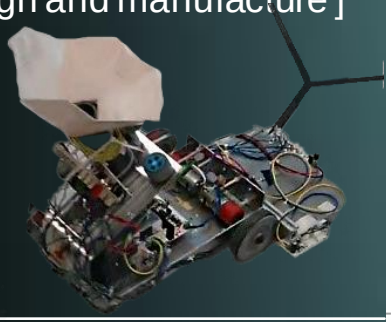
[Android, Java, BLE, GATT, product design, marketing]

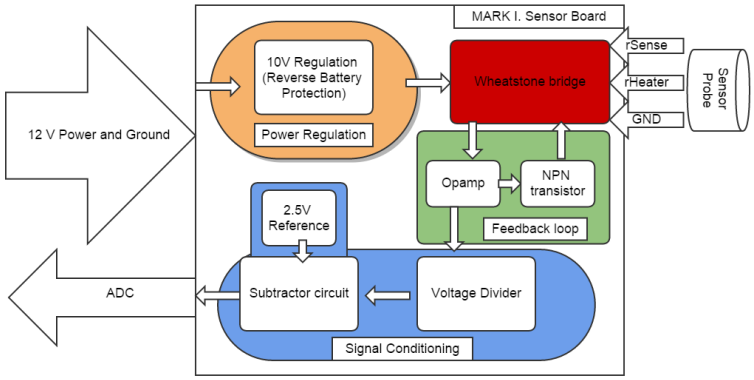
The purpose of the project was to design and develop an end-user product which is viable and marketable. As a team of five, we decided to create a wireless, low power smart bicycle health and fitness monitoring system. The key parameters relating to the cycling experience is gathered from various sensors placed on the bike (heart rate, pedal and cadence speed, rear proximity, elevation, temperature), each of them being equipped with its own microcontroller unit for extended configurability. The acquired data is then transmitted wirelessly via Bluetooth Low Energy from each module and presented in real time on the user's smartphone with the help of a dedicated "Smart Bike" Hub app for which I was responsible for. I had to design and implement the Android app as well as to realise the Generic Attribute Profile(GATT) server. The concept included a variety of unique features and a low price point to differentiate it from the competition, making it a prospective product.

Autonomous Robot Challenge

[Embedded C, Java, PID, PCB design and manufacture]

The team of four was required create a robot car which competed in following a line and performing additional tasks such as collecting and shooting table tennis balls within a time limit to win the race against other teams' designs. The given chassis, the two-wheel motors and two DC motors making up the shooting mechanism were complimented by our team with an H-bridge, an analogue reflection sensor, a fuse box circuitry and a Freedom Board (Kinetis KL25) by Freescale as the brain of the robot. The line following was implemented by me where I have calibrated the sensor and programmed the logic for PID control. I have also implemented a unique feature on the robot which was a WiFi Module that enabled remote controlling and update of PID constants over the air, thus facilitating the calibration process via a java app running on PC. The robot performed the challenges autonomously and was chosen to be in the top five designs.





Academic Work -- Key Highlights

Formula One Car Module

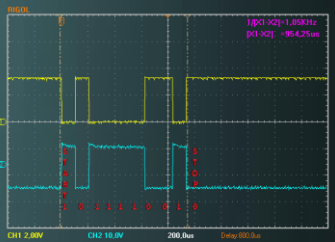
[CAN, product design, sensors prototyping, Wheatstone bridge]

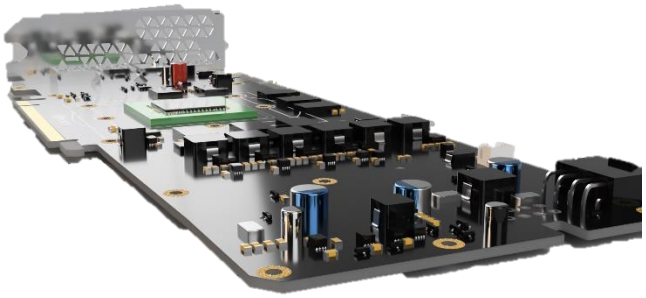
A three-man team had to build two units for the University of Glasgow's Student Formula One car: a sensor unit with Controller Area Network (CAN) protocol which measures and feeds the wind speed to the other modules on the car, including the second unit, which is the rear wing actuator. The wing actuator acts according to multiple parameters downloaded from the CAN bus such as breaking and the wind speed. I took charge of the sensor unit where I have researched about various sensing technologies and even created a DIY sensor probe from a tungsten filament and used a German tunnel for calibration. I considered multiple design choices regarding the sensor unit and created an IP 67 packaged solution made up of IST's thermal mass flow sensor and a Wheatstone bridge. I also exercised project management via GANT charts and scheduling. Making well-informed design choices, taking off some of the workload from my teammates and keeping the team in sync have resulted in the project's success.

Serial PI Comms

[Python, serial, oscilloscope]

The aims of the project were to establish serial connection between a Raspberry Pi and a PC, to package the data to ensure correct bundled information processing and to capture data trace being transmitted utilising oscilloscope and manually decode information. In order to establish serial connection using Python between the Raspberry Pi and the computer, the physical connections needed to be realised utilising USB-RS232 converters and RS-232 transceiver to account for different logic voltages. The python console application was written such that it can be as a server and a client too. In order to record and analyse the serial connection, a Rigol 50 MHz Digital Oscilloscope was used in combination to capture traces of on-going communication. The rate of erroneous transmissions was successfully reduced after implementing packetization in software and can be further enhanced by error detection algorithms.





Key Highlights -- DIY

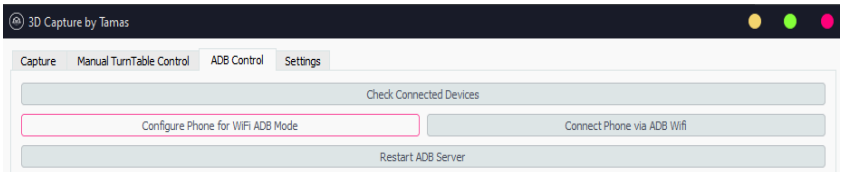
3D Design

[3D printing, Autodesk Fusion 360]

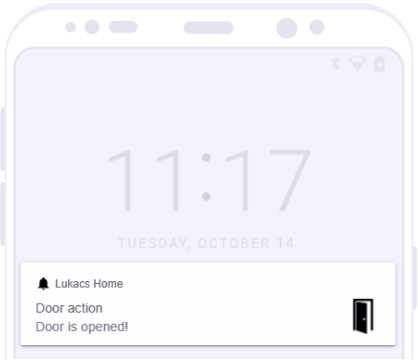
I have become acquainted with 3D designing though Autodesk's student design challenge and in which I succeeded and earned rewards utilising Fusion 360, now industry standard 3D tool. 3D design is now a valuable part of my skillset which opens up new possibilities plan and showcase, as well as staying up-to-date in the upcoming 3D printing era. Since the challenge, I have designed a number of items for prototyping, fixing of broken parts, personal satisfaction and even for autistic children in the form of development tools. My designs came into life by helping to assemble and utilise a DIY 3D home printer.

3D Capture Automatisation

[Python, PyQT5, subprocess, re, QThread, serial, Arduino]



I wanted to simplify, automate and integrate the processes involved with creating digital 3D files from scanning real world objects within one application. I had to design the API and program an Arduino to control a motorised turntable where the scannable subject would sit. Then I needed to gain control over the motorised turn table driven by the Arduino via serial connection to PC. Additionally, I had to gain control over an Android phone as photo source to take pictures and download them to the PC remotely using Android Debug Bridge (ADB).Finally, I had to initiate third party photogrammetry software though command line to post process images and output the desired 3D file. All of these sub-processes could be fine-tuned though the GUI interface of the tool with a tabular design. One of the most challenging pieces of the puzzle was controlling the phone remotely as it turned out to be harder than expected because file manipulation (grabbing latest image) is not well supported in Android via ADB. Additionally, timeouts needed to be handled carefully due to the asynchronous nature of the wireless phone control.



DIY -- Key Highlights

Push Notification Service

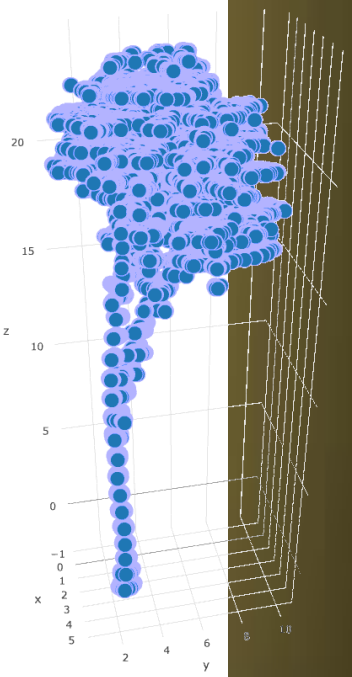
[Android, Java, Firebase]

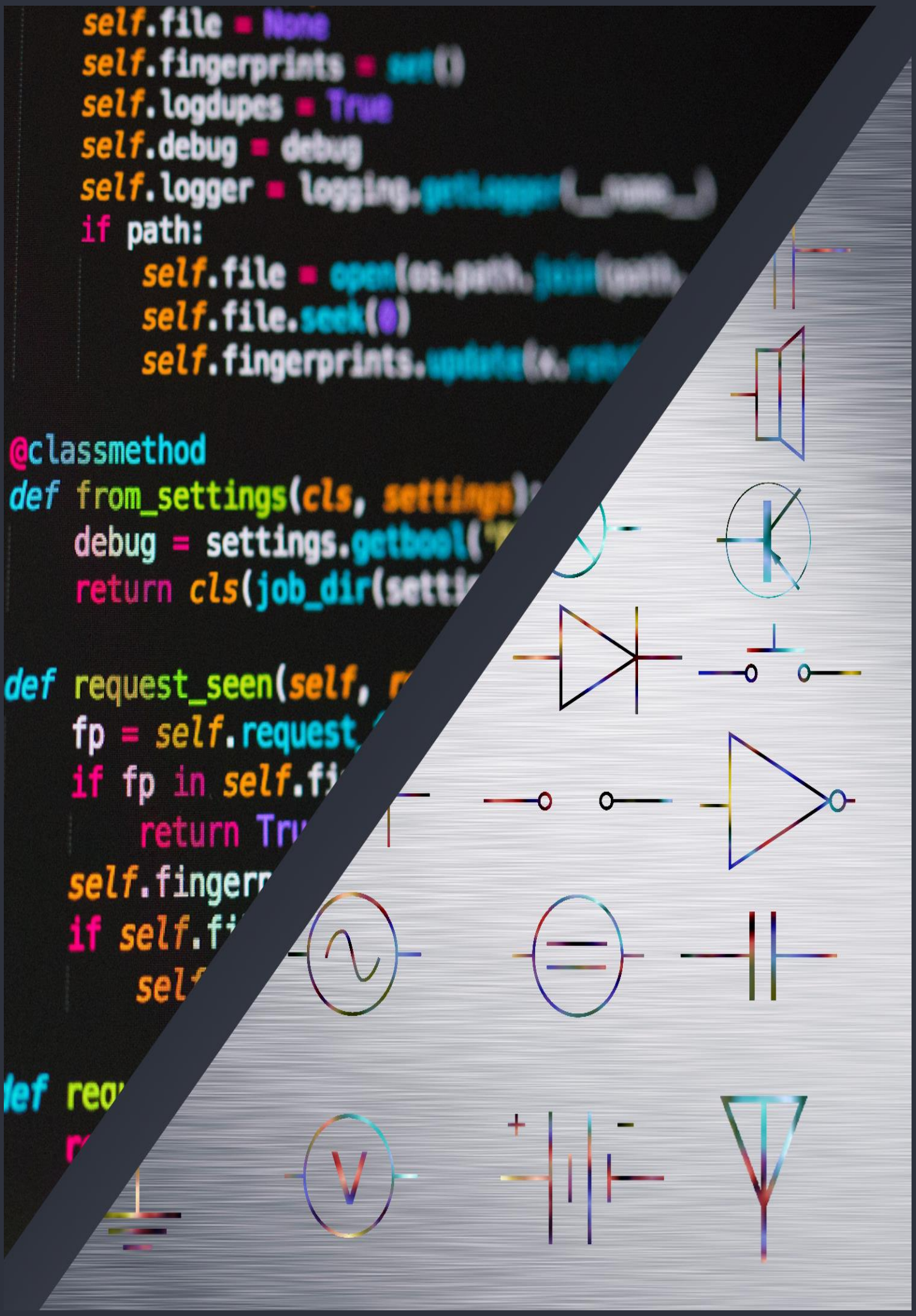
There was a need to be notified in case front door is opened. The sensor and microcontroller to monitor was already in place but the web service as well as client needed to be set up. I utilised Firebase Cloud Messaging server for push notification and implemented an Android application with a back-end service using Android Studio and Java to subscribe and receive any incoming notification sent out by the WiFi enabled microcontroller. One of the major hurdles of Android development is the constant flux and changes to the OS and to its development environment.

Tree Categoriser

[Python, numpy, plotly]

As part of a research project of a friend, scanned tree canopies had to be categorised based on shape. In order to jump start the project, I have introduced him to Python and explored how scanned data from the real world can be manipulated and converted into a point cloud format which then can be further processed. I have also explored how large datasets can be plotted as a 3D scatter plot utilising plotly package. I have furthered his insight and taught him of object-oriented approaches as well as how one can gather more statistics on the dataset utilising numpy.




self.file = None
self.fingerprints = set()
self.logdupes = True
self.debug = debug
if path:
@classmethod
def from_settings(cls,
return cls(job_dir(
def request_seen(self,
if fp in self.fi
self.fingerr
if self.fi