Lukasz Barczyk

Embedded Developer/Robotics Engineer

Address	London, UK
Mobile	(+44) 747 399 4492
Email	lukaszbarczyk@actuatedrobots.com
Website	https://actuatedrobots.com/
LinkedIn	www.linkedin.com/in/lukaszbarczyk

PROFESSIONAL EXPERIENCE

CEO & Founder – Actuated Robots Ltd (08/2022 - Now)

Providing firmware development, consultancy, embedded Proof-of-Concepts, trainings and workshops services

- Developed an embedded Proof-of-Concept solution of a remote-controlled, camera-based tracking system

  Technologies: *i.MXRT1062, AzureRTOS, protobuf, UDP-based communication network, GUI, reverse engineering*
- Created a free embedded programming and STM32F4 tutorial for educational purposes
- Currently developing projects such as motor controllers, automatic system testing platform, optical rotary sensors

Robotics Engineer – Agile Robots (10/2019 - 08/2022)

Leading firmware development for a robotic manipulator in the Robotics Unicorn Startup

- Developed Ethercat-synchronised firmware controlling motions on multicore microcontrollers from board bring-up, bootloading to reliable software (TI F28M36x Concerto)
- Coordinating firmware features and releases with other teams such as electronics, mechanics and motion control
- Initiated Continuous Integration testing ranging from on-host unit tests to on-target electronics integration tests
- Co-developed and advised on other projects such as multithreaded Linux-based point-to-point motion controller and communication stack, *Schmalz Vacuum Generator* gripper drivers and cable testing system
- Familiarity with industrial standards: IEC-61508(FuSa), ISO-10218(robot), IEC-60204(machinery), IEC-61784(networks)

Technologies: *TI F28M36x, STM32F0xx, C99, C++14/17, Python, Ethercat, Altium, Conan package management, CMake, Gitlab CI*

Embedded Firmware Engineer – Automata (04/2019 - 07/2019)

Safety-critical firmware for the 6-DoF manipulator

- Planned a customer-driven roadmap to achieve collision detection complying with cobot standard(ISO-15066)
- Implementing and comparing real-time collision detection algorithms on NXP KV5x microcontroller

Technologies: *NXP MKV58F1, CANopen, C99, FreeRTOS*

Robotics Engineer Intern – ZOA Robotics (11/2018 - 02/2019)

Embedded control software for a quadrupedal robot

- Implemented USB drivers for communication between Cortex-M4 and a Raspberry Pi
- Developed and implemented a control system for a test rig based on a jumping two-linked leg
- Tested and troubleshot BLDC motor drivers against heat emission

Technologies: *Cortex-M4, USB drivers, BLDC motor drivers, Raspberry Pi, gravity-compensated PD control*

Software Engineer – Ocado Technology (08/2015 - 07/2017)

Embedded control software for warehousing robots in a centralised multi-agent system

- Wrote and debugged C/C++ embedded software for STM32F4 microcontrollers controlling a mobile robot's motions
- Implemented sensor fusion from multiple analogue and digital sensors and analysed corner cases
- Wrote numerous unit tests using Catch framework, integration tests in Java and was introduced to HIL testing
- Rewrote about 160k lines of software written in C and C++ from two microcontrollers to one using QP framework
- Learnt event-driven programming, RTOS architecture and software patterns such as publish-subscribe
- Wrote a visualisation tool in python to analyse and make statistics based on movements of robots

Technologies: *STM32F4, SPI, UART, CAN, QP framework, FreeRTOS, Python, Java, Test-driven development*

EDUCATION

2017 - 2018 **MSc. in Robotics *(First Class Honours)***
King's College London

- *IEEE-RAS Regional Student Representative of the EMEA Region*
- Thesis: *Ultra-energy saving robot: Hopping MACCEPA-VD design*

2012 - 2015 **B.Eng. in Electrical and Electronic Engineering *(First Class Honours)***
University of Manchester

- *Team leader in the line-following buggy group project*
- Thesis: *Developing a Low Cost, Safe Manipulator for Fruit Harvesting*

PROJECTS

Hopping Legged Robot - Master's Thesis (2017 - 2018)

- Developed a hybrid dynamical model of a 2-link leg based on the Variable Stiffness Actuator with Variable Damping
- Implemented a dynamic simulation of the jumping leg in MATLAB to visualise the performance
- Implemented a PD controller for an in-flight stability
- Proved that variable damping can be exploited to harvest mechanical energy during jumping
- Designed in FreeCAD and built the physical implementation of the leg attached to a vertical stand
- Github Repository: *https://github.com/woookey/JumpingEnergyHarvester*

Event-driven, Real-time Framework for Modular Embedded Systems (2017 - 2018)

- Implemented an open-source framework for designing module-based embedded software in C from scratch
- Implemented RTOS mechanisms such as a memory manager, scheduler and intertask message passing
- Github Repository: *https://github.com/woookey/ModularSystemsFramework*

Robotic Arm - Bachelor's Thesis (2014 - 2015)

- Designed in SolidWorks and built two degree-of-freedom robotic arm to operate in its configuration space
- Simulated and compared PID and Computed Torque Control schemes in MATLAB
- Implemented CTC algorithm on the BeagleBone Black in C++

Line-following Buggy Group Project (2013 - 2014)

- Led a team of five students to build a line-following mobile robot
- Designed and tested electronic circuits such as an array of phototransistors and low-pass filters in Multisim
- Implemented interrupt-based PID control algorithm on a PIC microcontroller in C and PIC assembly

ADDITIONAL

- Attended multiple conferences such as IROS, ROSCon and IEEE-RAS Summer Schools where I presented my work
- Obtained certificates on robotics-related topics such as *Computational Motion Planning*, *Perception* or *Estimation and Learning* and many others (certificates available on LinkedIn)

SKILLS SUMMARY

Software	Eclipse, CLion, InteliJ, Altium Designer, Fusion 360, Multisim, SolidWorks, FreeCAD
Programming	Advanced (C, C++), Intermediate (Python), ROS, bash, PIC assembly software building (Make, CMake), QP Framework, unit testing (Unity, Catch), git, LaTex
Libraries & Algorithms	STM32Cube package, STL, Eigen; dynamic programming, genetic algorithms
Hardware & OSs	Windows, Linux, PIC, ARM and C2000 architectures, BeagleBone, Raspberry Pi, Arduino,
Languages	Polish (Native), English (Fluent), German (Beginner)

REFERENCES (*Available on request*)