Curt Henrichs
Software Engineer

✉ curthenrichs@gmail.com
📞 +1 (262) 422-7274
🌐 curthenrichs.github.io
github.com/curthenrichs
in linkedin.com/in/curt-henrichs
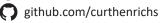📍 1402 Regent St. Apt. 604, Madison, WI 53711

Aspiration

I am passionate about building real things for real people.
My interests include human-robot interaction, embedded systems, software engineering, and mixed-reality interfaces.

Skills

Python, ROS, Git, Atmel, C/C++, Linux, Javascript, React, Angular, Unity/C#, Keras, MATLAB, MoveIt/Relaxed-IK

Education

(2018 - *) — **M.S. in Computer Science ~ Emphasis: HCI / HRI** — GPA: 3.7 / 4.0
📍 University of Wisconsin - Madison, Madison, WI

Coursework:

Human Computer Interaction	Artificial Intelligence	High Performance Computing
+ Wearables	+ Machine Learning	+ Adv. Computer Architecture
+ User Modeling	+ Computer Vision	
+ Data Visualization		

- Human Robot Interaction -

(2014 - 2018) — **B.S. in Computer Engineering ~ Emphasis: Embedded Systems** — GPA: 3.9 / 4.0
📍 Milwaukee School of Engineering, Milwaukee, WI

Coursework:

Embedded Systems	Software Development	Business / Management
+ Computer Architecture	+ Operating Systems	+ Servant Leadership
+ Digital / Analog Circuits	+ Data Structures	+ Entrepreneurship
+ Control Systems	+ Computer Vision	+ Ethics for Mgmt. and Eng.
+ Digital Signal Processing	+ Neural Networks	
+ Computer Networking	+ Computer Graphics	

- Engineering Practices -

Experience

(2019 - *) — **Graduate Research Assistant ~ People and Robots Lab, Computer Sciences**
📍 University of Wisconsin - Madison, Madison, WI

- Developed collaborative robot (cobot) authoring and training interfaces [Authr, Expert View Dashboard].
- Researched interactions with cobots for both attention management and levels of task interdependence.
- Worked with several colleagues outside of lab (in Human Factors and Optimization) to investigate cobot effectiveness when deployed on a variety of manual work activities.
  - Contributed robot capability analysis as inputs into allocation algorithm.
- Contributed to lab infrastructure and processes.
  - Ex. Maintained centralized robot description and configuration repository for lab.
  - Ex. Device bringup and documentation (Universal Robots UR3e, Microsoft Hololens 1 & 2).
  - Ex. Debugged and updated Robotiq gripper ROS drivers for colleague under paper deadline.
- Assisted colleagues with their user studies, technical development, and paper writing.

(2016 - 2018) — **R&D Software Engineering Intern**
📍 Dedicated Computing, Waukesha, WI

- Responsible for development of embedded firmware in C/C++.
- Prototyped server hardware and software systems [Matrix Storage, Fan Controller].
- Integrated embedded devices into server control software with Python.
- Developed internal hardware testing infrastructure [Thermal Chamber] with NodeJS, Python, and MongoDB.
- Contributed to product life-cycle documentation for design, implementation, and testing.

Notable Projects

🧪 Authr ~ Cobot authoring environment.
- Developed an Angular web app with ROS backend.
- Used MoveIt to compute motion plans and time-of-flight.
- Custom domain language designed around Therbligs.

🧪 Expert View Dashboard ~ Cobot training environment.
- React web app with ROS backend and Unity simulation.
- Operationalizes expert thinking into a checklist novices use to develop their programs with custom domain language.
- Explored Microsoft Hololens as an alternate XR interface.

🧪 pRAD ~ Evaluation of cobot operator interfaces.
- Applies predictive robot attention demand (pRAD) to cobots.
- Developed two widgets (timeline and timer) evaluated for task and collaborative outcomes in human-subjects study.
- Captured design suggestions from participant interviews informing future implementation.

🔲 Automated Thermal Chamber Testing.
- Developed several subsystems:
  - Unit-Under-Test state scraper captures CPU and GPU configuration / sensor values with NodeJS.
  - Thermal couple monitor running on NI cRIO.
  - Chamber control server, with NodeJS, issues low-level TCP byte commands.
- Integrated subsystems into internal testing software.

🔲 Matrix Storage ~ Server Backplane Controller.
- Developed controller firmware with Atmel C.
  - Aggregates PSU, fan, and environment sensing and control for Linux node on I2C system bus.
- Interfaced firmware with python application.

🔲 Programmable Fan Controller.
- USB UART with JSON API to configure programmable thermal profiles with individual fan control.
- Firmware written in C for Atmel ARM microcontroller.