

Swarup Majumder

swarupm2@illinois.edu | linkedin.com/in/swarup1 | swarupm3.github.io | 217-607-3792 | US Citizen

EDUCATION

University of Illinois at Urbana-Champaign <i>Bachelors of Science in Computer Engineering</i> Relevant Coursework: Data Structures and Algorithms (C++), Operating Systems (C, RISC-V, UART), FPGA Lab (SystemVerilog, AXI, SPI), GPU Parallel Programming (CUDA), Digital and Analog Signal Processing	GPA: 3.74/4.00 <i>Graduation: May 2027</i>
--	---

EXPERIENCE

Undergraduate Teaching Assistant <i>University of Illinois at Urbana-Champaign</i>	Aug 2025 – Present <i>Champaign, IL</i>
<ul style="list-style-type: none">• ECE 220: Computer Systems Programming<ul style="list-style-type: none">* Mentored 30+ students weekly, providing technical feedback for C, C++, and Assembly machine projects• ECE 385: Digital Systems (FPGA) Laboratory<ul style="list-style-type: none">* Assisted students with FPGA-based SystemVerilog designs with FSMs, VGA control, and SOC integration* Debugged timing, synthesis, and simulation issues using Vivado and on-board instrumentation	
Undergraduate Research Assistant <i>Parasol Lab, University of Illinois at Urbana-Champaign</i>	May 2025 – Present <i>Champaign, IL</i>
<ul style="list-style-type: none">• Benchmarked C++ STAPL and STL algorithm runtimes across 64 samples (99% Confidence Interval)• Enabled multi-GPU STAPL execution using C++ and CUDA, achieving 100x speedup over CPU baseline• Streamlined compilation, linking and execution for 25+ MPI configurations via Bash and Linux toolchains• Automated data visualization using Python to graph runtime metrics, accelerating algorithm analysis	
Robotics Software Mentor <i>FIRST Robotics Competition Team 4096</i>	Jun 2024 – Present <i>Champaign, IL</i>
<ul style="list-style-type: none">• Utilized Python APIs to interface with the CAN bus and reliably control 30+ motors across 8 subsystems• Authored structured Markdown technical documentation for software onboarding within a 40+ student team• Deployed computer vision AI object detection to provide operator feedback, reducing cycle times by 30%• Guided students in implementing a RUBIK-Pi based localization system, integrating camera-based pose estimation	

PROJECTS

Camera Vision Pipeline on FPGA <i>SystemVerilog, RTL Design, I2C, Vivado</i>	Nov 2025 – Dec 2025
<ul style="list-style-type: none">• Designed a pure RTL real-time camera pipeline with microsecond-scale frame latency on Spartan-7 FPGA• Initialized 40 camera registers via I2C protocol with a custom SystemVerilog controller for stable color output• Implemented filters and UV thresholding in SystemVerilog with line buffers for seamless edge detection• Integrated 7-bit Vivado XADC IP-based brightness dial with 128 discrete levels for precise luminance scaling	
GPT-2 Transformer Model <i>C++, CUDA, Bash, Linux, Slurm</i>	Oct 2025 – Dec 2025
<ul style="list-style-type: none">• Accelerated matmul and layernorm CUDA kernels with tensor cores and reduction trees, achieving 30x throughput• Built Bash sweep script to evaluate 80+ kernel configurations, optimizing inference throughput on Slurm clusters• Leveraged CUDA constant memory for weights/biases, reducing global memory traffic and forward pass latency	
Interactive Thermometer <i>Oscilloscope, Multimeter, Circuit Design</i>	Feb 2025 – May 2025
<ul style="list-style-type: none">• Built a temperature measurement circuit with op-amps and comparators, supporting a range of 0–99 °C• Designed an analog-to-digital stage to drive 7-segment displays showing temperature and user thresholds• Verified signal integrity, temperature, and threshold accuracy using an Oscilloscope and Multimeter	
24-Hour Clock <i>Arduino, C++, Circuit Design</i>	Feb 2025
<ul style="list-style-type: none">• Built an Arduino-based 24-hour clock using C++ to display time with precise minute-level accuracy• Implemented multiplexed control for 4x7-segment displays, reducing I/O to 11 pins with 4 ms cycles• Integrated a potentiometer interface converting analog input to time values for real-time calibration	

SKILLS

Programming: C, C++, SystemVerilog, Verilog, Python, CUDA, Bash, Java

Protocols/Interfaces: I2C, UART, SPI, CAN, AXI-4 Lite

Tools: Linux, git, gdb, Vivado, MPI, NVIDIA NSight, Slurm

Hardware/Equipment: Oscilloscope, Logic Analyzer, Function Generator, Multimeter, Soldering Iron