Monish Nene

Searching for job as Firmware or Software Engineer

☎ +1 (720) 688-6309 ✉ monish.nene@colorado.edu
https://www.linkedin.com/in/monishnene888a2412a/
Git Profile: https://github.com/monishnene

Education:

Masters in Embedded Systems Engineering University of Colorado, Boulder.	GPA: 3.4/4.0	May 2019
Bachelor of Engineering in Electronics VESIT, Mumbai University, Mumbai, India.	GPA: 8.00/10.00	May 2017

Skills:

Languages	Protocols	Platforms	Coding skills	Data Structures	Algorithms	Hardware	Development Boards
C	BLE	Git	String Manipulation	Trees	Sorting	Altium	Raspberry Pi (Linux)
C++	UDP	Linux	Big O Big Θ Big Ω	Linked Lists	Searching	Eagle	AT89C51 (Designed System on chip)
Python	UART	Keil	Data Logging	Hashmaps	Greedy	Logic port Analyzer	TI MSP432p401r (Baremetal)
GNU Make	I2C	Codeblocks	PID Control	Dictionaries	Recursion	Digital Oscilloscope	ARM cortex A8 (ARM Architecture)
Shell/Bash	SPI	Matlab	State Machines	Stacks	Backtracking	Motors	ARM cortex M0+, M4, (Baremetal)
ARM assembly	Regex	Jupyter	Real time pthread scheduling	Queues	Dynamic Programming	Memory Mapping	Cypress PSoC 5LP CY8C5868AXI-LP035
8051 assembly	Bit Banging	QT5	Object Oriented Programming	Graphs	Divide & Conquer	Interfacing Sensors	Silicon Labs EFR32 Blue Gecko BGM 121

Experience:

- **Graduate Teaching Assistant for the course Embedding Sensors and Actuators** *August 2018-Present*

The course gives hands-on experience on converting analog signals from various sensors to computable data and controlling the speed, position and torque of different types of motors. PID control, analog & digital filters, sensor working principles and calibration.

Projects:

- **HTTP-based web server** *[Ubuntu(Linux)]{C}* *October 2018*
- **File Transfer using UDP protocol** *[Ubuntu(Linux)]{C}* *September 2018*
  - Reliable data transmission using acknowledge for packets. Uploaded 12 MB file on a remote server.
- **Time-lapse** *[Raspberry Pi (Linux) using openCV library and camera]{C++}* *August 2018*
  - Frames were captured at 1Hz and saved as in ppm with embedded timestamp and jpeg format. 5 Real-Time threads were used for scheduler, capture image, save as ppm, edit ppm header and compress and save in JPEG format
- **Customized ' .bashrc' file** *[Linux]{Shell / Bash}* *May 2018*
- **Data Transmission via Optical Medium** *[TI MSP432p401r (Transmitter) and AT89C51(Receiver)] {C}* *April 2018*
  - Audio signal converted to digital signal with ADC and sent over optical medium using Laser and Photodiode. Audio signal at 160kbps sent as data with a custom asynchronous communication with start bit and data byte. Bit-Banging done at the receiver.
- **Accelerometer Controlled Robot - Bluetooth low energy /** *Silicon Labs EFR32 Blue Gecko BGM 121* **/** *{C}* *April 2018*
  - Complete Bluetooth Protocol written in C. Client developed with software event scheduler to handle 2 Services at a time. Accelerometer was on the handheld client and according to its orientation, the command was sent to the server. The server moved a chassis according to command from client. Encryption Decryption using AES32 and MITM protection done.
- **System on chip Design for AT89C51** *{C,8051 Assembly}* *March 2018*
  - Power, Reset, Clock, NVSRAM interfacing, SPLD for memory mapping, RS232 Connection with UART, Bootloader mode switch, Debug Latch, 16 x 4 LCD Interfacing, EEPROM with I2C interface, DAC controlled with SPI, user interface designing.
- **Health Thermometer - Bluetooth Low Energy /** *Silicon Labs EFR32 Blue Gecko BGM 121* **/** *{C}* *February 2018*
  - Sleep routine to keep the average current below 5uA.Tranmission power adjusted with feedback. Over the air update successful.
- **Laser Detector and follower /** *Cypress PSoC 5LP CY8C5868AXI-LP035* **/** *{C}* *December 2017*
  - A LASER was attached to a stepper motor moving +10° to -10°. Accelerometer feedback for initial homing. 2 LDRs attached to a linear slider DC motor. The slider moved according to feedback from the LDRs and followed the LASER using a state machine. Feedback from a Potentiometer on slider used to avoid ramming at the ends. The system worked at max 3Hz frequency
- **Interfacing SPI, Data Logging, DMA and Profiling /** *Beagle Bone Black (Linux), FRDM KL25Z] {C}* *November 2017*
- **Voice Controlled Robot** *[Raspberry Pi (Linux) , Arduino]{Python}* *March 2017*

Courses: Principles of Embedded Software, Real Time Embedded Systems, Embedded Systems Architecture, IoT Embedded Firmware, Design and Analysis of Algorithms, Embedding Sensors and Actuators. Network Systems, Embedded System Design,