

# Monish Nene

Searching for job as Firmware or Embedded System Engineer

+1 (720) 688-6309 ✉ [monish.nene@colorado.edu](mailto:monish.nene@colorado.edu)

<https://www.linkedin.com/in/monishnene888a2412a/>

Git Profile: <https://github.com/monishnene>

## Education:

Masters in Embedded Systems Engineering (2017 – 2019) University of Colorado, Boulder.

(GPA: 3.34)

Bachelor of Engineering in Electronics (2013 – 2017) VESIT, Mumbai University, Mumbai, India.

(GPA: 3.50)

## Technical Skills:

Scripting and Software Development Languages: C, Python, ARM Assembly, 8051 Assembly, MATLAB.

Software and Hardware Tools: Digital Signal Oscilloscope, Logic Port Analyzer, Version Control, Altium, Firmware IDEs.

Other Skills: Hardware and Software Debugging, Component Evaluation & Selection, Signal Conditioning, Dhrystone Benchmarking.

## Course work:

Principles of Embedded Software

Embedded System Design

Real Time Embedded Systems

IoT Embedded Firmware

Embedding Sensors and Actuators

Mastering Embedded Systems Architecture

## Projects:

- **Data Transmission via Optical Medium** [TI MSP432p401r (Transmitter) and AT89C51(Receiver)] {C} *April 2018*
  - ( video link - <https://www.linkedin.com/feed/update/urn:li:activity:6396109419922411520/> )
  - 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 it's orientation, the command was sent to the server. The server moved a chassis according to command from client. The second service was a feedback from the robot to the client about the ambient light measured with a I2C interfaced LUX sensor. 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*
  - Server Developed to Measure Temperature and send data to a mobile application Client. Code was developed with sleep routine to keep the average current below 5uA. Transmission power adjusted with feedback from Client. Over the air update compatibility.
- **Laser Detector and follower** [Cypress PSoC 5LP CY8C5868AXI-LP035] {C} *December 2017*
  - ( video link - <https://www.linkedin.com/feed/update/urn:li:activity:6359496404951449600/> )
  - 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*
  - Task queue and interrupt handlers used for scheduling. DMA used for data memory movements. Binary data logger with RTC time stamp to log encoded data for tasks performed. The logs were later decoded using python and converted to text. Profiling, finding the execution time of functions done for various parameters. Interfaced a Nordic Wireless transceiver with SPI protocol.
- **DC Motor Torque and Speed Control** [Cypress PSoC 5LP CY8C5868AXI-LP035] {C} *October 2017*
  - Speed control using PWM and Torque Control by controlling armature current. Current was controlled by using Power MOSFET in linear range or by changing frequency of PWM. Feedbacks used were Encoder, Back EMF and Current Ripple separately. PID Control was used. Analog and Digital Filtering done to remove noise from feedback and get a smooth result.
- **Interfacing UART** [ FRDM KL25Z (Arm Cortex M0+ Bare metal OS) ] {C} *October 2017*
  - Created circular buffer data structure for receive and transmit buffers. Code for UART protocol written. Designed a user interface to give input via UART. Interrupt handlers and software event scheduler used. Documentation done using Doxygen software.
- **Linux Compilation and Cross-Compilation** [Ubuntu(Linux), Beagle Bone Black, FRDM KL25Z] {C} *September 2017*
  - Wrote datatype conversion and memory management functions along with error checks. Compile time switch macro used to make the code compatible for all platforms. Created a Makefile to select the compiler according to platform, compile C files, link required object files and build an executable file. Unit tests on functions done using CMOCKA and projects test code.
- **Voice Controlled Robot** [Raspberry Pi (Linux) , Arduino]{Python} *March 2017*
  - Voice was converted to a string offline using CMU Sphinx library on Raspberry Pi. Strings were converted to commands and were sent to an Arduino via HC- 05 Bluetooth module. The Arduino moved the robot chassis according to the command received.