Mrunal Prakash Gavali

ROBOTICS ENGINEER

✉ gli.mrunal@gmail.com ☎ 8879519722 https://github.com/Mrunal-G

EDUCATION

Bachelor's Degree
Robotics Engineering
University of Central Lancashire
GPA : 81%
2017 – 2020

SKILLS & TOOLS

Python
C#
ROS
Embedded C/C++
RTOS
ARM STM32 Cortex M4
Keil SDK IDE
STM32CUBE IDE
Arduino
Sensor Integration
Proteus
KiCad
Linux
Git/Github
SQLite
MATLAB

COURSES

Robotics and Autonomous Systems
Embedded Real-time Systems
Micro-controller systems
Machine Intelligence
Software Development 2
Data Communications
Robotic Systems
Instrumentation and Control
Electronic System Applications
Digital Systems

OTHER EXPERIENCE

President of UCLan Engineering Society | 2018-19

Student contributor to Google funded SenseMaker project - a joint collaboration between UCLan and the Machester Evening | 2019

EXPERIENCE

Engineering URIP Undergraduate Summer Research intern

University of Central Lancashire, United Kingdom| 06/1/2018 –10/31/2018

- Researched on various electric vehicles converter technologies, especially the Triple active Bridge (TAB) DC-DC converter and Dual Active Bridge DC-DC converter and embedded electronic devices within electric vehicular (EV) networks as well as charging infrastructure designed for EV vehicles.
- Implemented hardware models using MATLAB and Simulink to calculate the power losses in MOSFET switches of the EV battery packs.

PROJECTS

Implementation of Genetic Algorithm to find optimal engine configuration project

Project dealt with designing, implementing and testing a genetic algorithm program from scratch in C# to find the optimal engine configurations by evaluating the fitness score of each chromosome string in the population after selection, crossover, and mutation operations of the optimal chromosome strings at each iteration.

Multi-threaded robot interface for mobile robot communicating using Xbee module

Mobile robot project involving creating a real-time user interface in C# using computer vision and multi-threading techniques (backgroundWorker threads) for a small mobile robot communicating with the PC using Xbee module. Computer vision techniques are used to extract red/blue objects from real-time video using filters for object detection and tracking.

Developed a smart boxing glove IoT prototype on Cortex-M4 NUCLEO board and visualized the sensor data in C# GUI with SQLite database

Created a smart boxing glove prototype (sensor integration) using sensors like heart sensor TCRT 1000, temperature sensor, accelerometer, SD card for boxers on STM32L476RG ARM board for the purpose of monitoring the biometric of the boxer like heart beats, body temperature, punch counter, etc., on the GUI. The desktop application (GUI) in C# also allows to log the details of the boxer in SQLite Database. IoT programming for sensors with STM32L476RG Cortex-M4 ARM board is done using Mbed/Keil SDK. The graphical user interface (GUI) in created in visual studio. Ecad software Proteus is used for rapid prototyping .

Implementing a primary smart home IoT prototype on Cortex-M4 NUCLEO board for elderly and disabled people using Mbed

Created a home automation prototype using sensors and actuators for elderly and disabled people on STM32F303RE ARM board for the purpose of monitoring emergencies like falls or seizures.

Developed finite state machine to mimic a simple car parking machine Cortex-M4 microcontroller

Developed a finite state machine (FSM) to mimic the operation of a simple car parking ticket machine on ARM microcontroller STM32L476RG using C programming language in Keil IDE. Unit testing is implemented during embedded software development.

Developed signal capturing system to capture 1000 samples from any signal generated from the oscilloscope in register-based C and LL Libraries with Cortex-M4 ARM board in Keil.

The developed system captures varying voltages in the range 0-3.3V using ADC and displays signal characteristics like minimum and maximum RMS, voltage and frequency on the serial terminal using UASRT2 to communicate with the PC via the ST-Link Virtual COM Port. ADC conversion is implemented through program and the output (push-pull for DAC_OUT2) is displayed again into oscilloscope using DAC.