ICT & Robotics (Open-Learning)

**Student**: Shallwin Eleathon Silvania

**PCN**: 494141

# Introduction

This self-made track is meant to build on the Technology Track. In this track we focus on the “nerves” if you will of modern-day. Physics & math are a must for this self-made track seeing that you will be working with simulations & algorithms.

## Learning outcomes

The track is divided into two modules the Simulation module and the Application module. Logically you must complete the Simulation Module before you can start working on the Application Module.

|  |  |
| --- | --- |
| **ICT & Robotics:** | **Descriptions:** |
| Robotics Fundamentals | I research the fundamentals of robots. |
| MATLAB & Simulink Fundamentals | The fundamentals of programming in MATLAB and basic Simulink usage. |
| C++ Programming Fundamentals | The fundamentals of programming in C++. |
| ROS Fundamentals | The fundamentals of the ROS2 |
| **Elective\*** | X |
| Final project: A rover or drone | I will create a small drone which will have bits of everything I learned this semester. |
|  |  |
| **Open Choice List:** |  |
| Robotics Motion Planning | Analysis on existing Motion Planning algorithms. |
| Robotics Perception (Computer Vision) | Analysis on Computer Vision. |
| Robotics Communication (IoT) | Analysis on Communication Protocols |

Robotics Fundamentals

* I understand what it means to be a robot
* I understand the dynamics of a robot
* I can analyze the workings of a robot

MATLAB & Simulink Fundamentals

* I can declare & use (built-in & self-defined) functions in MATLAB
* I can control the flow in a MATLAB script
* I can use different loop statement in MATLAB
* I can work with Array, Vector & Matrices in MATLAB
* I can plot data in 2D & 3D using MATLAB (Animation)
* I can create models in Simulink
* I can run simulations using models (built/imported) in Simulink
* I can visualize output data of a model
* I can use MATLAB in Simulink

C++ Programming Fundamental

* Recap C programming (Tech S2) – Variable, function, control flow, loops etc.
* I can design C++ software using UML
* I can make Use Cases for my software
* I can write Unit Tests for my software
* I can practice OOP using C++
* I can work with Header Files
* I can work with pointers
* I can work with dynamic memory

ROS Fundamentals (C++ & Python)

* I understand & can setup ROS2 Publishers
* I understand & can setup ROS2 Subscribers
* I understand & can setup ROS2 Nodes
* I understand & can setup ROS2 Server Services
* I understand & can setup ROS2 Parameters
* I understand & can setup my own custom interface for ROS2
* I can create & configure ROS2 launch files

Final Project: A Rover

* A simulation is made to show the workings of the robot
* Robot must be controllable
* Robot must be able to detect obstacles using 2 or more different sensors
* Robot must be able to traverse using obstacle detection data

**Electives:**

Robotics Motion Planning

* I can implement various MP algorithms in MATLAB/C++
* I can run simulations using MP algorithms
* I can work with actuators

Robotics perception

* I can analyze & use different sensors
* I can use data from sensors to construct digital maps
* I can work with OpenCV

Robotics communication

* I understand what a communication protocol is
* I can use communication protocol (such as Bluetooth, Wi-Fi etc.)
* I can analyze and choose the right CP for my applications

## Planning

|  |  |
| --- | --- |
| Week 1: | Robotics Fundamentals |
| Week 2: | Robotics Fundamentals |
| Week 3: | MATLAB & Simulink Fundamentals |
| Week 4: | MATLAB & Simulink Fundamentals |
| Week 5: | MATLAB & Simulink Fundamentals |
| Week 6: | C++ Programming Fundamentals |
| Week 7: | C++ Programming Fundamentals |
| Week 8: | C++ Programming Fundamentals |
| Week 9: | ROS Fundamentals (Robot Operating System) |
| Week 10: | ROS Fundamentals (Robot Operating System) |
| Week 11: | ROS Fundamentals (Robot Operating System) |
| Week 12: | **Elective\*** |
| Week 13: | **Elective\*** |
| Week 14: | **Elective\*** |
| Week 15: | Final Project |
| Week 16: | Final Project |
| Week 17: | Final Project |
| Week 18: | Final Project |
| Week 19: | X |
| Week 20: | X |

## Challenges

These are the challenges that need to be completed to pass this made-up course. Before starting the challenges, one must take the courses given below to have a solid foundation on the topics.

**Background Information:**

* [**https://www.udemy.com/course/matlab4b/**](https://www.udemy.com/course/matlab4b/)
* [**https://www.udemy.com/course/beginning-c-plus-plus-programming/**](https://www.udemy.com/course/beginning-c-plus-plus-programming/)
* [**https://www.udemy.com/course/ros2-for-beginners/**](https://www.udemy.com/course/ros2-for-beginners/)

*Robotics Fundamentals:*

Write a research paper on the different part that make a robot and compare an existing robot with what you researched.

* Min 3 Pages of information
* IMRAD structure
* APA stye referencing

*MATLAB & Simulink Fundamentals*

* Challenge 1: Research 3 Sorting Algorithm and implement them in MATLAB code
  + A document explaining how the algorithms work
  + The MATLAB scripts that implement these algorithms
* Challenge 2: Import a datasheet and visualize the data
  + The datasheet
  + The MATLAB script that produces the visualization
  + A document explaining what the visualizations mean
* Challenge 3: Simulate movements using a Robot Arm template
  + A document of the model before and after modification, with explanations on the modifications.
  + The modified model & MATLAB scripts (if one is written)

*C++ Programming Fundamentals* (following TECH S3 guidelines)

* Challenge 1: Make a library for a sensor (OOP)
  + A document explaining what the sensors do and what your library does.
  + The library code (CPP & Header)
* Challenge 2: Linked List (C programming Challenge TECH S3)
  + The all the code that implements linked list functionalities.
  + Working proof of implementation
* Challenge 3: Digitalization of Hardware
  + A document with a description of the Hardware you are building
  + A UML Design of the application
  + Unit-Test for all the classes
  + Code for the application

*ROS Fundamentals*

* Challenge 1: ROS2 setup
  + A document explaining how you did it (with proof that you did it)
* Challenge 2: Node Communication
  + A document explaining what is being communicated
  + All files (Both C++ & Python version)
* Challenge 3: TurtleBot simulation
  + A document explaining what movement you trying to recreate
  + The code written to achieve the movement (in C++ or Python)

***Electives:***

*Robotics Motion Planning*

* Challenge 1: Motion planning Algorithms (based on a given map)
* Challenge 2: Motor control

*Robotics Perception (Computer Vision)*

* Challenge 1: Sensor data processing and digital map creation (Arduino or Raspberry Pi)
* Challenge 2: OpenCV Object detection

*Robotics Communication (IoT)*

* Challenge 1: Routing algorithms (based on a given map)
* Challenge 2: Peer to Peer communication