**Title: Finite Difference Time-Domain Method Implementation in C++**

By: Xhoni Robo

Supervisor: Prof. Dr. Peter Thoma

Co-Supervisor: Prof. Dr. Egbert Falkenberg

This is a short and crude unofficial description of the topic for my Master Thesis, to clear up any common misconceptions on my (the student’s) side and come up with possible improvements along the way.

The basic idea of this thesis is to create a functional implementation of the Finite Difference Time-Domain Method in C++, which will be a standalone open source console (or GUI if time allows for this) application. This will start off as a modest implementation at first, allowing to simulate 2D scenarios. Then the plan is to implement 3D scenarios and make it so that the application uses the GPU for processing instead of the CPU, which will allow for faster processing speeds altogether. This will be done through CUDA or QUDA, which are open source libraries in C++ that allow the implementation of this functionality. For starters, I will use my personal laptop device which has a Nvidia 2070 RTX GPU which should be powerful enough to handle this simulation. If time allows it, we can later make this implementation remotely accessible and place it in a more powerful computer to allow for even greater calculations. The implementation and writing of the thesis will follow an Agile Development Plan. Below is a simple version explanation on how I plan to complete this thesis:

**Total time:** 5 Months (20 weeks total)

**Sprint Frequency:** Weekly or Bi-weekly, with a short call with the main supervisor at the end

**Total Sprints:** 10 – 20 sprints

**Plan:**

1. Collection of information regarding FDTD (2 weeks)
2. Begin implementation and prepare development environment (1 week)
3. Implement 2D scenario simulation (3 weeks)
4. Implement 3D scenario simulation (3 weeks)
5. Implement GPU processing (3 weeks)
6. Run some test scenarios (1 week)
7. Discuss and implement further features (5 weeks)
8. Prepare final documentation and presentation (2 weeks)

**Total:** 2 + 1 + 3 + 3 + 3 + 1 + 5 + 2 = 20