THE UNIVERSITY OF TENNESSEE KNOXVILLE

Neuromorphic Autonomous Racing - Preliminary Design Report

Submitted December 15th, 2022

Team 3

Thomas Neuefeind, Kellen Leland, William Johnson, Bryson Gullett, Robert Schaffer, Mahim Mathur

Sponsored by Dr. Catherine Schuman

COSC 401 Senior Design

Team Signatures and Names

Thomas Neuefeind	Kellen Leland	William Johnson
Bryson Gullett	Robert Schaffer	Mahim Mathur

EXECUTIVE SUMMARY

The purpose of utilizing neuromorphic algorithms for autonomous racing – through the application of spiking neural networks — is to demonstrate the feasibility of neuromorphic solutions for autonomous driving such that a vehicle can safely map obstacles, optimize speed while reducing barriers such as weaving, and discern interactions in the environment.

The concept of a neuromorphic autonomous car is extremely practical, extends to real-world applications, and is a subject of interest in the advancement of neuromorphic solutions. The goal is to build a one-tenth scale race car and track based on other autonomous race cars and explore software to facilitate and optimize driving to meet this feasibility. The race car will act as a testing and research platform for the TENNLab neuromorphic research group. This team also aims to provide a well documented and accessible platform that can be used as an entry point for those interested in neuromorphic computing.

The top five design concepts Team 3 had to consider were: power, input sensing, signal processing, neuroprocessor implementation, and software to train spiking neural networks. Regarding hardware, this team will use components recommended by the F1TENTH website to build the autonomous race car. However, some of the hardware suggested by F1TENTH was either out of stock or did not fit well into the neuromorphic system the team plans to build. Consequently, after thorough design concept evaluation, the team ultimately decided to select alternative parts pertaining to power, input sensing, signal processing, and control. For instance, the team decided to use a larger battery pack for longer battery life and better compatibility, the Hokuyo 10LX LiDAR for its sampling rate and cost, and a Raspberry Pi 4 to process input signals for its reasonable but not overkill computing performance. Also, TENNLab already has an FPGA neuroprocessor implementation ready to run the spiking neural networks the team trains. Furthermore, this team will be using the already-existing software base from both TENNLab and the F1TENTH project to simulate and train the race car.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

PROBLEM DEFINITION & BACKGROUND

Neuromorphic computing uses networks of simulated neurons and synapses to model a system that operates similarly to a human brain. These networks are called spiking neural networks (SNNs), and they run on a special type of processor called a neuroprocessor. There are relatively few implementations demonstrating the practicality of applying these networks to real world situations which can be attributed to a lack of accessibility to the needed hardware and software. Dr. Schuman and TENNLab have been striving towards developing effective frameworks for training the models needed to create suitable platforms and workflows that will foster a greater adoption of spiking neural networks and advance neuromorphic solutions.

Neuromorphic computer design offers many advantages to the traditional von Neuman architecture where the CPU is separate from the memory. Neuromorphic computers use the synapses and neurons to control both the memory and processing [1]. They are highly parallel, since the neurons and synapses operate simultaneously, and scalable, since the neuromorphic chips can be chained together to increase the number of neurons available [1]. Neuromorphic computers also use drastically less power during operation since much of the spiking neural network is idle with just a few neurons and synapses doing the work at any time. These properties make neuromorphic computers an ideal candidate for many applications, enabling continued performance improvements and addressing issues such as the looming end of Moore's law [1]. A summary of these differences between standard von Neumann computing versus neuromorphic computing is shown below in Figure 1.

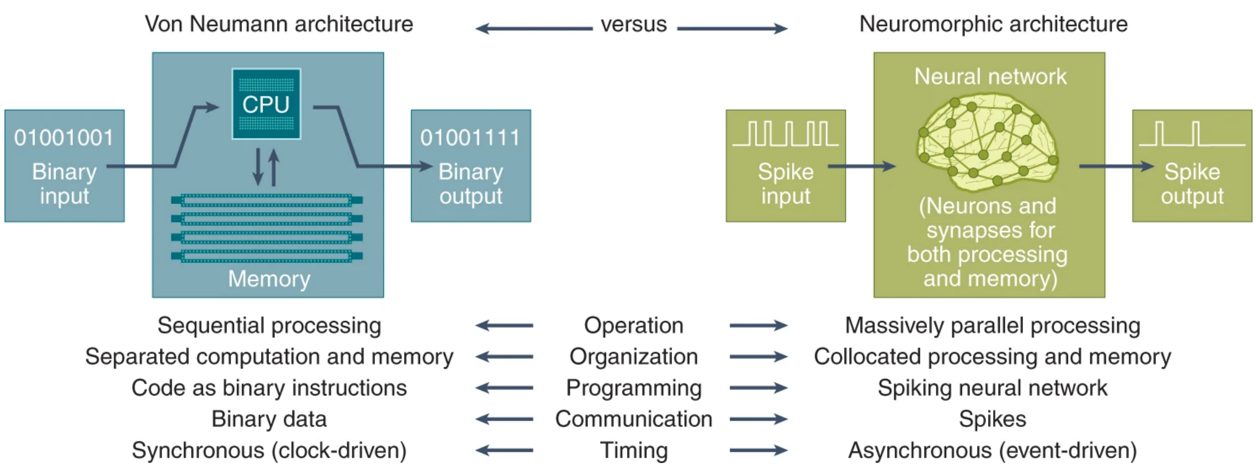


Figure 1. Comparison of Von Neumann and Neuromorphic Architectures

Source: Adapted from [1]

The team has been tasked with engineering a one-tenth scale model race car that uses a LiDAR sensor to detect the surroundings and a spiking neural network to make decisions that will be used to drive the car around various race tracks. Dr. Schuman has previously manufactured a similar car with ORNL and TENNLab and has requested the development of another car to continue her research at the University of Tennessee. The car will be used for further research, applications, and testing by the TENNLab group in the future.

As part of this project, the group will be evaluating the sensor design on the cars, investigating different training approaches and measures of success, and deploying and evaluating neuromorphic solutions on the physical car. The team will be engineering solutions to problems Dr. Schuman encountered in her research with the car. Primarily, the team will work on increasing the limited speed that was achieved by reducing the excessive wheel turning that occurred during testing. A communication bottleneck, that occurs due to the large amount of data received from the LiDAR that needs to be processed by the Raspberry Pi and communicated to the neuroprocessor while receiving the outputs from the neuroprocessor, will need to be evaluated for any possible improvements. At the end of the project, the team will race another team working on a different car.

REQUIREMENT SPECIFICATIONS

The customer is Team 3's sponsor, Dr. Schuman, and the major requirements can be broken down to the following table of customer requirements:

Table 1. Customer Requirements

1. Build the race car using the components in the bill of materials listed in Budget
2. Use spiking neural networks to train the race car with the following goals in mind a. Optimize the race car b. Increase the speed incrementally with the desire of 15 miles per hour
3. Fix communication overheads between software and hardware
4. Meet up with another team working on this same project to ensure that racing with two race cars will be compatible
5. Demonstrate the project by racing this race car with the other team's race car

In detail, the requirements are to build a race car and to program software that makes the race car autonomous. The main objectives for this race car include moving through obstacle courses autonomously and winning in a race against another team – Team 28 – that is making their own race car. With these objectives in mind, additional requirements that come up are that the car needs to be able to avoid obstacles, and so this team – Team 3 – will need to test the race car to avoid hitting the other team's car during the race planned for the final milestone. For the obstacle courses, each track needs to be unique so that this team can test that the race car generalizes well against many tracks, and does not just overly learn only one track. A successful run is defined to be the completion of two laps in a given track with no collisions.

Another aspect of this project is to have the car move through various tracks as quickly as possible. After the team's discussion with Dr. Schuman, she has made it clear that the goal for this race car is to achieve 15 miles per hour while racing. This proposed speed is because in past iterations of this project, the highest speed the car went was 10 miles per hour, and the current objective is to extend this speed. Other components of the speed requirement include optimizing the car to take the shortest path it can find, making the car drive as straight as possible to increase speed, and minimizing weaving. For the weaving specifically, the last race car that Dr. Schuman worked with had a noticeable weaving motion while driving, and she tasked this team with investigating ways to reduce it.

Additionally, Dr. Schuman stated that with her last race car, there was a latency issue with respect to the communication between the software that runs the car and the car itself. One of this team's tasks is to investigate a more efficient way for the race car and the software to communicate. This task will also aid in increasing the speed of the car while racing because the car will then make calculations for desired paths more efficiently.

The greatest risk identified is hardware failure because there are many considerations to take into account such as proper assembly of the race car, storage of the race car and its components, usage of the safety bags for charging the car's battery, and testing of the race car on low speeds to mitigate collisions before deploying the car to test at higher speeds. To ensure that this risk will be mitigated, a team member must be accompanied by at least one other team member during any of these procedures.

DELIVERABLES

The primary deliverable of this project is providing the customer with a single one-tenth scale electric race car that is controlled autonomously using a neuromorphic system. The car the team delivers should be fast enough to have a top speed close to 15 miles per hour on straightaways and should not have obvious extraneous, jittery turning movements. The car will be delivered with the following components:

Table 2. Car Component Deliverables

1.	Chassis and wheels
2.	Electric motor and turning servo
3.	Motor/Speed controller
4.	Neuroprocessor to run the spiking neural network that controls the car
5.	Processor(s) to process data that goes into and comes out of the spiking neural network
6.	LiDAR sensor
7.	Battery

In addition to just the race car, the team must also deliver a race with the other senior design team working on the exact same project. The race should include both cars simultaneously competing on the same track to determine the effectiveness of each team's design. As an additional challenge, cars should avoid contact during the race as well.

Also, the team plans to deliver the software and spiking neural networks used to control the race car. Because Dr. Schuman's work is primarily focused on neuromorphic software development for applications and spiking neural network training, she will want to review the team's work in these areas. The software the team plans to deliver will consist of an application that utilizes the TENNLab's existing neuromorphic framework and a simulator built for F1TENTH cars [2]. This application will be designed to train the spiking neural network(s) that controls the race car.

All of these deliverables should be provided to the customer before the end of the spring 2023 semester.

PROJECT MANAGEMENT / MILESTONES

Responsibilities

Each team member will be delegated different responsibilities as they become necessary throughout the life of the project. Jira will be used to track overall progress, assign tasks, create issues, and unify documents in a central repository. The roles designated to each member will be a focus but will not be exclusive. Each member will be actively involved in every aspect of the project and will be assigned a wide range of tasks. Team members will be expected to be available to help with major or high priority tasks even if assigned to another team member.

Kellen Leland will act as the Team Leader and Reviewer for the project. Weekly scrum meetings will be scheduled to share progress updates and discuss roadblocks. Kellen will act as scrum master through the first sprint, but this role will also be shared so every team member can gain this valuable experience. Kellen will also be responsible for updating and maintaining the team's Jira website and coordinating meetings with other team members or the project sponsor.

There are two solutions architects – one for software and one for hardware – assigned to Thomas Neuefeind (Computer Scientist) and Bryson Gullett (Computer Engineer) , respectively. In addition, Thomas Neuefeind will be the lead researcher and Bryson Gullett will be the lead implementer for the project. The Solutions Architects will work directly with the Team Leader in creating and distributing all tasks that arise throughout the life of the project.

The rest of the team – Mahim Mathur, William Johnson, and Robert Schaffer – will be Designers. The designers are responsible for working directly with the Team Leader and Solutions Architects to design and implement both hardware and software solutions. The designers will also be responsible for working with the rest of the team to rate the difficulty and assign the necessary time needed to complete each task. Mahim will additionally fulfill the role of Librarian and will help to record everything that is discussed in both the team's weekly scrums and meetings with the team's project sponsor. Specifically, Mahim will keep track of each version of the project and update the project requirements for each given week that is tasked to each team member.

On top of these roles, each team member will be a tester. Testing is a critical part of the design and implementation process. Every team member will be responsible for helping to test each other's hardware and software solutions to help verify correctness and integrity before they are integrated into production software or hardware products. Dr. Schuman's lab will house the race car and tracks, and will be used for the majority of the testing.

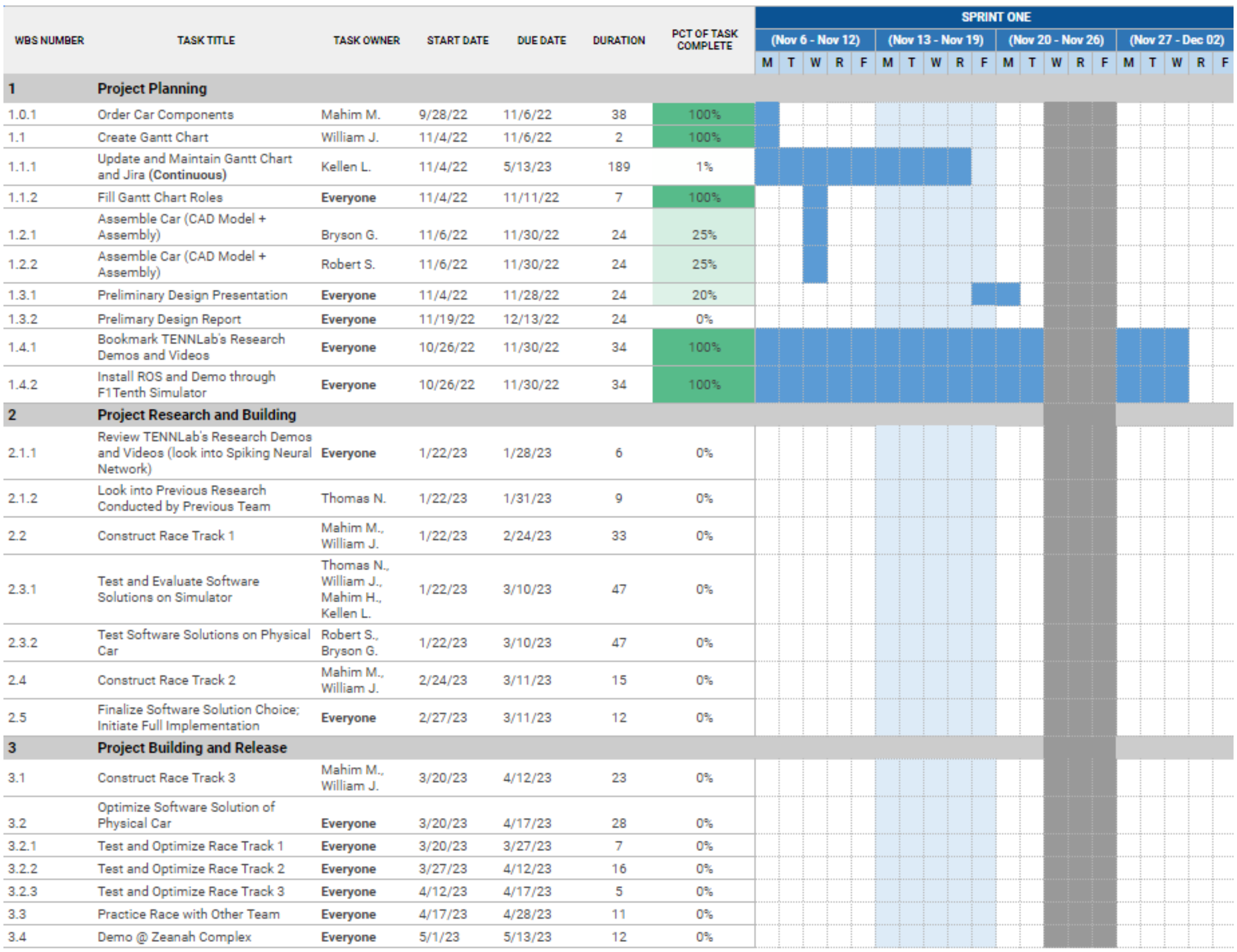
Milestones

Project management and tracking will use the agile software engineering model and there will be a total of three sprints. Table 3 below details each of these three sprints.

Table 3. Project Management Sprint Table

Sprint 1	**Date range: 09/28/2022 - 01/22/2023**
The first sprint will end on January 22nd, 2023 and will focus on the initial spin up of the project. By the end of the first sprint this team expects all parts to be ordered and received, car assembly to be completed, at least one track to be assembled with common materials like cardboard, and verification and demonstration of software control of the physical car to be functional. The team also expects to have a firm grasp of the use of the F1TENTH simulator and TENNLab neuromorphic framework.	
Sprint 2	**Date range: 01/22/2023 - 03/10/2023**
The second sprint will end on March 10th, 2023 and will focus on the software aspect of the project. During this sprint, the team expects each member to gain familiarity with the software stack in order to test and evaluate different software solutions on the provided simulator, and to decide on a handful of software solutions to initiate testing on the physical race car. Construction of the first and second track will be fully assembled during this sprint to prepare for testing a race in Sprint 3.	
Sprint 3	**Date range: 03/12/2023 - 05/13/2022**
The third sprint will end on a date that is to be determined as this team will need to communicate with Team 28 that is also working on this same project to establish the final race date. The focus of this sprint will be to finalize the software solution, optimize the hardware and software for racing on multiple unique tracks, and prepare the car for the race against the other team for the project demo day. Third track will be assembled after testing the first track, and will be tested after optimization is fully assessed for the first and second tracks.	

To better visualize project management milestones and individual tasks, this team has created a Gantt chart. Below is a snippet of the Gantt chart. The full chart contains all three of the sprints mentioned above.



WBS NUMBER	TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION	PCT OF TASK COMPLETE
1	**Project Planning**					
1.0.1	Order Car Components	Mahim M.	9/28/22	11/6/22	38	100%
1.1	Create Gantt Chart	William J.	11/4/22	11/6/22	2	100%
1.1.1	Update and Maintain Gantt Chart and Jira **(Continuous)**	Kellen L.	11/4/22	5/13/23	189	1%
1.1.2	Fill Gantt Chart Roles	**Everyone**	11/4/22	11/11/22	7	100%
1.2.1	Assemble Car (CAD Model + Assembly)	Bryson G.	11/6/22	11/30/22	24	25%
1.2.2	Assemble Car (CAD Model + Assembly)	Robert S.	11/6/22	11/30/22	24	25%
1.3.1	Preliminary Design Presentation	**Everyone**	11/4/22	11/28/22	24	20%
1.3.2	Prelimary Design Report	**Everyone**	11/19/22	12/13/22	24	0%
1.4.1	Bookmark TENNLab's Research Demos and Videos	**Everyone**	10/26/22	11/30/22	34	100%
1.4.2	Install ROS and Demo through F1Tenth Simulator	**Everyone**	10/26/22	11/30/22	34	100%
2	**Project Research and Building**					
2.1.1	Review TENNLab's Research Demos and Videos (look into Spiking Neural Network)	**Everyone**	1/22/23	1/28/23	6	0%
2.1.2	Look into Previous Research Conducted by Previous Team	Thomas N.	1/22/23	1/31/23	9	0%
2.2	Construct Race Track 1	Mahim M., William J.	1/22/23	2/24/23	33	0%
2.3.1	Test and Evaluate Software Solutions on Simulator	Thomas N., William J., Mahim H., Kellen L.	1/22/23	3/10/23	47	0%
2.3.2	Test Software Solutions on Physical Car	Robert S., Bryson G.	1/22/23	3/10/23	47	0%
2.4	Construct Race Track 2	Mahim M., William J.	2/24/23	3/11/23	15	0%
2.5	Finalize Software Solution Choice; Initiate Full Implementation	**Everyone**	2/27/23	3/11/23	12	0%
3	**Project Building and Release**					
3.1	Construct Race Track 3	Mahim M., William J.	3/20/23	4/12/23	23	0%
3.2	Optimize Software Solution of Physical Car	**Everyone**	3/20/23	4/17/23	28	0%
3.2.1	Test and Optimize Race Track 1	**Everyone**	3/20/23	3/27/23	7	0%
3.2.2	Test and Optimize Race Track 2	**Everyone**	3/27/23	4/12/23	16	0%
3.2.3	Test and Optimize Race Track 3	**Everyone**	4/12/23	4/17/23	5	0%
3.3	Practice Race with Other Team	**Everyone**	4/17/23	4/28/23	11	0%
3.4	Demo @ Zeanah Complex	**Everyone**	5/1/23	5/13/23	12	0%

Figure 2. Project Management - Gantt Chart

BUDGET

This team's budget is primarily aimed at hardware components for building and testing the race car. No software budget is required as the university – TENNLab's Neuromorphic platform and computing resources such as ISAAC – provides these services for students and researchers.

Dr. Schuman's Budget

The budget given to Team 3 by Dr. Schuman is $6,000, but the expenditure of materials for this project is roughly $3,000. This team's bill of materials is based on F1TENTH's master bill of materials for the Traxxas Slash and the Traxxas Rally race cars which fall within this $3,000 price range [3]. The following major car components are included in the bill of materials, along with their costs:

Table 4. Major Car Components and Prices

Chassis ($423.99)	**Fasteners ($111.73)**	**Sensors ($1,595.00)**	**Battery ($309.88)**	**Powerboard ($161.02 - *composed of several units*)**
Cables/Connectors ($47.76)	**Compute Module ($192.99)**			

Included in these major components are the following sub-components:

- **Chassis**
  - Traxxas Slash chassis model *
  - Platform deck acrylic sheet
- **Fasteners**
  - M3 standoffs 6mm hex (with lengths of 14 mm, 25 mm, and 45 mm)
  - M2.5 x 0.45 mm, 10 mm long FF standoffs
- **Sensors**
  - LiDAR sensor (Hokuyo 10LX) *

- **Speed Controller**
  - FLIPSKY FSESC 6.7 PRO Speed Controller *
- **Battery**
  - LiPo Battery (and Charger)
  - Two LiPo safety backs for charge and storage
- **Powerboard**
  - LTC3119 Voltage Regulator and TPS565201 Voltage Regulator
  - Inductors, capacitors, and resistors for various voltages, tolerance, and capacitance
  - Other components – toggle switch, barrel jack, terminal blocks, and balancer port, LEDs
- **Cables/Connectors**
  - USB C to Pigtail
  - TRX to XT90 Adapter
  - Traxxas ID Charge Lead Adapter
  - VESC PPM Cable
- **Compute Module**
  - Raspberry Pi 4, 4 GB RAM (with a 512 GB microSD card) *

* represents components that contain **a unit** that cost $200 or more; these units are to be delicately handled

The four critical components are the chassis, the LiDAR sensor, the speed controller, and the compute module. Of these four components, the LiDAR sensor is the most critical component since it is the most expensive component in the bill of materials and it plays a major role in detecting obstacles. While there are cheaper LiDAR sensors, the Hokuyo 10LX is recommended as it has reliable measurement data and a 270 degree field-of-view. The LiDAR sensor will be used in this project to sense the track walls and provide input to the spiking neural network for control processing.

Cross-referencing F1TENTH's bill of materials, the team thoroughly explored the exact car components in late September as well as similar car components to compare pricing and view the availability of each component and subcomponent listed above. Components were gathered from different vendors such as Amazon, Digikey, McMaster-Carr, Flipsky, Traxxas, and Walmart, and were compiled into the team's bill of materials.

As shipment time is one of the challenges with ordering parts, the intended car construction date of late November to early December had to be pushed back to early January. Every component should arrive by that date, so that the car can be built as soon as possible to be fully evaluated, trained, and tested.

Course Budget

While there is some leftover budget allowance of a little less than $3,000 in Dr. Schuman's budget after the expenditure of this team's car components are finalized, there is no immediate plan to use this leftover allowance. Instead, the course budget will be allocated for spare parts like extra powerboard components or standoffs, additional hardware such as a remote-controlled kill switch in case software-implementation is infeasible, and emergencies such as replacements for damaged parts. Of course, every team member will be responsible for ensuring that no parts are damaged in the process of testing the race car.

Ultimately, the objective is to minimize spending of the course budget as much as possible to plan ahead for failure; that is, the course budget serves as emergency funding to accommodate for unexpected failures during implementation and testing of the autonomous race car. And in the absolute worst case scenario, the remaining budget from Dr. Schuman will be taken into consideration.

Timeline of Budget and Shipment of Parts

September 28th, 2022
The team arrived in the Min Kao balcony to discuss parts selection and to build the Bill of Materials list. In the Bill of Materials, team members visited vendors online such as Traxxas, McMaster, FLIPSKY, and DigiKey, and compared quality and price to determine the parts' fit to the Bill of Materials list.

October 8th, 2022
The team finalized the Bill of Materials and sent this list over to the project sponsor Dr. Schuman to be processed. Due to having many vendors, shipment times for the various parts displayed variance.

October 19th, 2022
The first wave of parts arrived, with some parts including sub-components to cables and connectors, powerboard components, and the LiPo battery. There remains more sub-components to arrive for these major components.

Meanwhile, Bryson and Robert met up with Dr. Schumann for the key to the lab where the assembly of the car will take place. Since there were not enough parts to construct the car yet, they continued their design on the CAD model for the car to plan out the assembly.

November 10th, 2022
The team explored a different vendor for the chassis since Traxxas announced supply-chain

issues incurring a delay on delivery. Bryson selected the same chassis from a different vendor with the same price as the chassis listed on Traxxas, the Traxxas Slash 4x4 VXL - Brushless.

November 15th, 2022
The second wave of parts arrived, with the arrival of the speed controller. In addition, more sub-components to the powerboard arrived.

November 30th, 2022
The third wave of parts arrived, which included the Hokuyo 10LX LiDAR sensor and the Traxxas chassis. The remaining components should arrive between early December and no later than early January.

TECHNICAL APPROACH

This project can be roughly broken down into hardware and software components. These components both require that various design decisions be made. Thanks to previous research, many of the hardware design decisions have been narrowed down. For example, this project will require some type of input sensor to eventually feed data into the neuroprocessor. The work done previously has made it clear that a LiDAR is a good choice for such a sensor. However, since there are many LiDARs on the market, the team still must choose which is the most appropriate for this project. The main issue facing design decisions concerning hardware is that many parts are out of stock. As such, replacement parts with similar performance have been evaluated.

The design space of the software is still very open. In order to further the team's progress here, the team must first become familiar with the simulators provided by F1TENTH. That being said, the software side of this project will involve writing an application that utilizes F1TENTH's autonomous race car simulator as a machine learning environment in order to train a spiking neural network that can control the car. This application will be written in Python and use extensive neuromorphic frameworks provided by the TENNLab. Furthermore, the team will use TENNLab's EONS neuromorphic evolutionary machine learning algorithm to run the application and train the spiking neural networks.

The only decision that can be made so far in the software space is a quite simple one: F1TENTH provides two simulators. One simulator is easier and quicker to use, but the other simulator supports multi-agent systems, or in other words, more than one race car. However, this project requires racing against another race car. Once the first basic spiking neural networks are trained on the simulator, the team will be faced with many more design choices regarding the software on the car in order to optimize performance.

The software components require further research and work to be better defined. However, it is possible now to functionally decompose the hardware, as shown in Figure 3 on the next page.

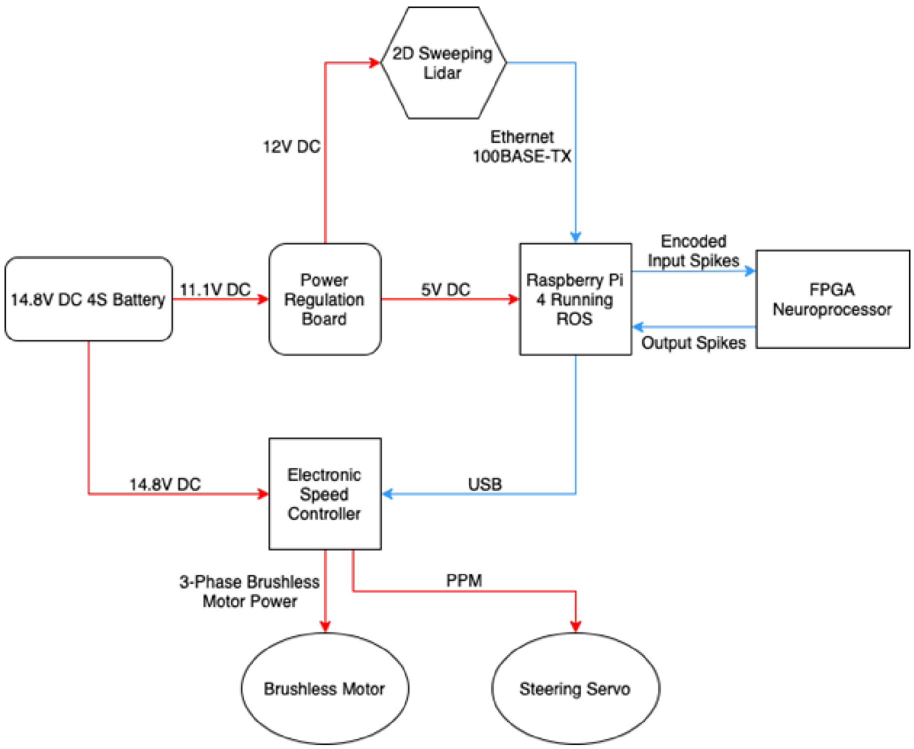


Figure 3. Hardware Interfacing Block Diagram

From the block diagram above, three main subsystems present themselves: power, input, output, and processing. It can be seen that there are two main types of connections between these subsystems. The data connections are represented in blue, while power signals are in red. Further, the labels on the arrows indicate the type of connections and interfaces between the components.

The block diagram details some of the design decisions that had to be made for the hardware components of this project. Though some changes needed to be made to the parts listed in the F1TENTH build due to part shortages and neuromorphic system incompatibilities, many of the components were readily available. This allowed the team to focus on the four primary hardware design decisions that were inconsistent with the F1TENTH project: power, input sensing, signal processing, and neuroprocessor implementation. Out of these four component decisions, the two major design decisions regarded the sensory input and signal processing.

Regarding sensory input, the autonomous race car must be able to sense the track walls in order to know when to accelerate, decelerate, and steer. As discussed before, previous work indicates that LiDAR will likely be the best option of input sensor; however, there are a large variety of LiDARs on the market that detect object distance and location to solve this problem. Resultantly, the team had to decide on which of these sensor products to purchase.

The other major design decision pertains to signal processing. This team's build is primarily different from the F1TENTH build in that Team 3 will be using an FPGA neuroprocessor to run a spiking neural network that controls the car. Therefore, not a lot of parallel computing power will be required in the auxiliary computing unit (Raspberry Pi above), as it will be used for signal processing by encoding and decoding spikes from the FPGA. However, the chosen signal processing component still needs to run ROS to nicely interface with the LiDAR input and speed controller.

In summary, the team focused its technical design efforts this semester on making design decisions regarding the sensory input and signal processing subcomponents and integrating them into the rest of the F1TENTH autonomous race car build. In particular, it was important to choose the best LiDAR and signal processing unit for this particular application. It should also be noted that future software design decisions will be focused solely on improving the trained performance of the spiking neural network running on the FPGA. Therefore, interfacing the FPGA with the input sensor and output devices via an auxiliary compute unit should be efficient and fast.

DESIGN CONCEPTS, EVALUATION AND SELECTION

From a software standpoint, the team easily decided to use F1TENTH's multi-agent simulator combined with TENNLab's extensive neuromorphic frameworks and machine learning algorithms; however, design decisions were much more interesting when it came to the hardware. Most of the team's hardware design is based on the previously existing F1TENTH project. However, due to out of stock items and the fact that this team's car will be controlled by a neuroprocessor, a few hardware changes had to be made from the F1TENTH build. Specifically, the team had to make hardware design decisions regarding the neuromorphic processor implementation, input sensing, signal processing, and power. Two of these design decisions were rather straightforward. TENNLab already has a neuroprocessor implementation running on an FPGA that Dr. Schuman would like the team to use. The other simple design decision revolves around power; a DC battery whose voltage and current ratings depend upon the electronic speed controller is required. Therefore, the team chose a 14.8V DC four cell battery due to its compatibility with F1TENTH's suggested speed controller.

There were then two major design decisions left to settle this semester that drastically differ from the F1TENTH framework. The first of these decisions is deciding which LiDAR (Laser imaging, Detection, and Ranging) module the team wanted to use as the sensing component of the autonomous race car. The team's race car needs a LiDAR with a decent range (approximately 10 feet) to sense turns in the track ahead and a high scanning frequency (approximately 100 Hz) to get fresh data to the fast-moving system as quickly as possible. Therefore, three unique options for this component (Yujin YRL2-05, Hokuyo UST-20LX, and Hokuyo 10LX) were considered and graded based on their cost, scanning range and maximum horizontal angle, scanning frequency, and the ease of integration. The weighted decision matrix can be seen in Table 5 below.

Table 5. Weighted Decision Matrix - LiDAR

LiDARs	Costs	Scanning Range & Horizontal Angle	Scanning Frequency	Ease of Integration	Totals
Weights	*1*	*3*	*5*	*4*	
Yujin YRL2-05	5	2	1	2	24
Hokuyo UST-20LX	1	4	4	5	53
Hokuyo 10LX	4	4	4	5	**56**

From this table, the Hokuyo 10LX was determined to be the best option for the build. This LiDAR possesses all of the features the team was looking for and is a sufficient middle ground between the two other choices. In contrast, the Hokuyo UST-20LX was more feature rich than what the team really needed, and the Yujin YRL2-05 did not have the required scanning frequency or range. The Hokuyo UST-20LX has a scan detection range of 20 meters and the team only needs around 10 meters of range for the car - which is the range on the Hokuyo 10LX. The Yujin YRL2-05 only has a detection range of 5 meters, which was found to be insufficient for our build. Both of the Hokuyo options have a scanning frequency of 40 Hz, where as the Yujin can only handle up to 20 Hz. The extra costs of the Hokuyo UST-20LX was found to not be worth it as the main difference is the extended range which the team does not need. The Yujin YRL2-05, while cheaper, did not meet the necessary specifications. The Hokuyo 10LX was found to be the most optimal choice when balancing costs and features.

The second major design decision needed was to determine the compute module to use for signal processing. The LiDAR sensor values need to be encoded into a form that the FPGA neuroprocessor can understand and manipulate. Similarly, the output values need to be decoded and processed into signals that can be sent to the electronic speed controller for use in controlling the brushless motor and steering servos to drive the car. Several requirements for the signal processing compute module were considered. First, the processor must be able to run the Robot Operating System (ROS), which recommends a 1.6 GHz processor and 2 GB of memory. Second, it must have sufficient processing power and memory to be able to handle both encoding the raw input values into spikes for the spiking neural network and decoding the encoded output values while the car is in motion. The weighted decision matrix for the compute module for signal processing can be seen in Table 6 below.

Table 6. Weighted Decision Matrix - Compute Module for Signal Processing

Processors	**Costs**	**Processor Speed**	**Memory**	**Ease of Integration**	**Totals**
Weights	*1*	*6*	*3*	*4*	
Raspberry Pi 3 (1200 MHz w/ 1 GB RAM)	3	2	3	4	40
Raspberry Pi 4 (1.5 GHz w/ 4 GB RAM)	2	5	5	4	**45**
Arduino Nano (16 MHz w/ 4 KB RAM)	5	1	1	2	22

These three compute modules (Raspberry Pi 3, Raspberry Pi 4, and Arduino Nano) were considered for the team's build. The compute modules were graded based on their cost, processor speed, memory, and ease of integration with the other components. The Arduino Nano, while less expensive than the other options, proved to not be powerful enough to run the

Robot Operating System (ROS) or handle the processing of both input and output data simultaneously. The Raspberry Pi 3 and 4 both proved to be comparable options, but the team decided that the extra cost for the Raspberry Pi 4 would be the most effective for the extra processing power and RAM. These extra compute resources could become beneficial for running ROS well and encoding and decoding spike values for the spiking neural network as quickly as possible.

After extensive design concept evaluation, the team decided to use the Hokuyo 10LX LiDAR input sensor due to its high sampling rate and reasonable cost, and it chose to use the Raspberry Pi 4 because it could run ROS with high performance. Consequently, these components may be placed in Figure 3 in the Technical Approach section to provide a final subsystem breakdown of the neuromorphic autonomous race car. In summary, the overall design concept for the solution includes: an RC car chassis, a 14.8V DC four cell battery, a power regulation board, the Hokuyo 10LX LiDAR sensor, a Raspberry Pi 4 for signal processing, an FPGA neuroprocessor implementation provided by TENNLab, an electronic speed controller, a brushless drive motor, a servo steering motor, and software provided by both TENNLab and the F1TENTH project for training a spiking neural network to control the car.

REFERENCES

[1] C. D. Schuman, S. R. Kulkarni, M. Parsa, J. P. Mitchell, P. Date, and B. Kay, “Opportunities for neuromorphic computing algorithms and applications,” *Nature Computational Science*, vol. 2, no. 1, pp. 10–19, Jan. 2022.

[2] “F1TENTH — Course Documentation,” f1tenth.org. https://f1tenth.org/learn.html

[3] “Bill of Materials — F1TENTH - Build latest documentation,” *f1tenth.readthedocs.io*. https://f1tenth.readthedocs.io/en/stable/getting_started/build_car/bom.html