A Novel Docking and Communications System for Heterogeneous Modular Robots

James White

P.h.D

School of Physics, Engineering and Technology
University of York

September 2023

Abstract

Modular robots are systems of many independent modules that can mechanically join and communicate with one another to form a robot morphology that can accomplish more than any of the modules could accomplish independently. Systems of modular robots that can dynamically reconfigure can form an ideal morphology for a given task, or repair the system if one or more modules fail. Such capabilities are useful for applications such as operating in harsh environments and exploration of extra-terrestrial planets. Previous work in the field has been focused on locomotion strategies, reconfiguration strategies and docking connector design. While these are all very important areas of research, if modular robots are to progress to the stage of being deployable systems, work needs to be done on developing a holistic platform that research can be built upon. This thesis describes the work that has been done to create a new heterogeneous cubic modular robot platform, Mo*, with particular focus on developing a system that can be used as a standard for modular robot development through a novel connection mechanism and communication strategy It is hoped that this can advance the field of research by allowing providing a platform that can be used for a wide range of future research The system is designed and built, with experiments being performed to test the performance of the docking mechanism and communication system between modules. The hardware system developed in this work could be used and built upon to further advance the capabilities modular robotic systems for real world applications.

Table of contents

List of figures

List of tables

Declaration

I declare that the research described in this thesis is original work, which I undertook at the University of York during 2019 - 2023. This work has not previously been presented for an award at this, or any other, University. Some parts of this thesis have been published in conference proceedings; where items were published jointly with collaborators, the author of this thesis is responsible for the material presented here. For each published item the primary author is the first listed author

White, J., Post, M.A. and Tyrrell, A., 2022, July. A Novel Connection Mechanism for Dynamically Reconfigurable Modular Robots. In International Conference on Informatics in Control, Automation and Robotics (ICINCO) 2022: proceedings. IEEE.

James White
September 2023

Acknowledgements

I would like to express my gratitude to my supervisors Mark Post and Andy Tyrrell for providing guidance, support and freedom to explore the topic during the course of my research.

I would also like to thank my PhD colleagues in the department for many informal conversations on various aspects of my research, as well as the friendship and comradery enjoyed throughout my time at York.

Finally I would like to thank my partner, Mary, for enduring me throughout the course of my research. You helped me navigate a lot of stressful times and were a dependable listener and councillor throughout the highs and lows.

Chapter 1

Introduction

From when they were first invented, robots have greatly impacted the world. The Cambridge dictionary defines a robot as "a machine controlled by a computer that is used to perform jobs automatically" [1]. Machines that are capable of reliably and repeatedly performing tasks that would otherwise be undertaken by a human operator has transformed sectors such as manufacturing. In recent years there has been significant development in autonomous robots, where an expert operator is not required to give explicit commands to a robotic system in order for it to perform tasks.

Robotic systems to date have typically been monolithic, where the robot is composed of one highly interconnected system. While these systems have been the major development area in robotics, they have one major downfall. If one component in the robot breaks, the whole system is most likely rendered inoperable. Expert intervention (system designer/engineer) is required to fix the broken part(s) before the robot can be operational again.

The field of modular robots aims to address this issue by constructing robotic systems out of discrete sub-systems called modules, which on there own are not a useful robotic system, but by mechanically connecting together through standardised interfaces can construct a variety of useful systems. If one part of a module breaks, then that module can be easily replaced without the need for complex disassembly, diagnosis and reassembly operations.

The field of robotic systems can be thought of as a spectrum. On one side you have completely centralised monolithic systems, and on the other side there is swarm robot systems, which are completely decentralised. Modular robotics lies somewhere between the two of these and can tend towards more monolithic or more decentralised. The rational behind this is that neither monolithic nor truly decentralised systems have proven to be capable of the wide range of activities that are desirable for them to perform, so a 'best of both worlds' approach is taken.

The first modular robotic system to be developed was CEBOT [2] by Fukuda et. al. at Nagoya University in Japan. The authors cited benefits of modular systems such as self-organisation (the ability of modules to organise themselves in to useful structures), self-evolution (the ability to adapt to varying tasks and environments), and functional amplification (gaining multiple functional modes from one set of hardware).

1.1 Motivation

While there has been much research in to modular robotics in recent years for varying applications [3–10], the research has yet to produce a system that can be deployed for real world applications outside of a research laboratory setting. Moves towards creating a usable modular robotic system in the real world could enable areas such as space exploration and search and rescue to advance greatly, where re-configurability and in-situ self repair are crucial. It could be said that the applications of modular robots are only limited by the imagination of the system designer.

1.2 Problem Definition

There has been much work to advance the field of modular robotics since the first system was introduced in 1990 by Fukuda et. al. Most of the research though has been focused on small, incremental advances to certain aspects of a modular robot system such as docking connectors, reconfiguration algorithms, locomotion and distributed control. With each of these research projects, researchers have developed their own system that is tailored to the specific improvement that they are seeking, leaving the field with lots of modular robotic platforms that are good at one specific task. This has resulted in a lack of standardisation in modular robotics, failing one of the main objectives of the research field

1.3 Aim and Objectives

The main aim of this research project is to create a modular robotic platform standard that is designed to be usable and extensible for a wide variety of applications. There is no particular target application of this system, rather that it could be the groundwork for a system that can be used to create modular robotic systems that can come out of research laboratories and in to real world applications.

The aims and objectives of this work can be summarised as follows:

- To design and construct a modular robotic platform framework that is adaptable and extensible to a wide variety of applications.
- To design and construct a connection mechanism to enable the modules in the system to mechanically connect to each other.
- To design and implement a communication strategy for modular robot systems that can handle the types of data throughput that modern sensors and processors require.

1.4 Contributions

The contributions of this thesis are:

- A novel modular robotic platform standard called Mo*, which includes a 10 cm cubic module that is constructed of sub-modular tiles that interface with each other through a simple, standardised connection allowing the design of modules with extended capabilities to be faster, which will in turn reduce the time to getting modular robots used in real world applications.
- A novel genderless magnetic connection mechanism for modular robots that only requires power upon transition from connected to unconnected states. This connection mechanism also integrates an electrical connection between modules through the same magnets that are used for the connection coupling force between modules. The connection mechanism is capable of performing single sided disconnection, allowing a system to eject a broken module that cannot operate its own connection mechanism.
- A communication strategy for modular robot platforms which creates a step-change in the capabilities of communication systems in modular robots. This system implements 100Mb/s communication between modules, on a packet switched network that is robust to reconfiguration at run time. The system uses the IEEE 802.3 100Base-T Ethernet standard for the physical layer, with minor modifications, and a new compressed network protocol that is tailored for modular robotic systems.
- A simulation platform designed to provide complete modelling and evaluation capability for the Mo* modular robot platform. This simulator implements physics simulation

using Unity engine, a network simulator for the communication between modules, and simulation of various aspects of the electronic subsystems.

1.5 Publications

The following paper on some of the work presented in this thesis was published during the course of the research:

- **White, J.**, Post, M.A. and Tyrrell, A., 2022, July. A Novel Connection Mechanism for Dynamically Reconfigurable Modular Robots. In International Conference on Informatics in Control, Automation and Robotics (ICINCO) 2022: proceedings. IEEE.

During the course of the research, the author also contributed to other projects not directly related to this thesis. These have led to the publication:

- Gorma, W., Post, M.A., **White, J**., Gardner, J., Luo, Y., Kim, J., Mitchell, P.D., Morozs, N., Wright, M. and Xiao, Q., 2021. Development of modular bio-inspired autonomous underwater vehicle for close subsea asset inspection. Applied Sciences, 11(12), p.5401.

1.6 Thesis Outline

This thesis is presented with the following structure:

- **Chapter 2** contains a review of the literature of modular robotics to date. This provides context for the thesis and identifies the areas of the field which are yet to be explored. A quantitative review of connection mechanisms and communication in modular robots is performed, followed by a detailed review of particular notable systems. After this a detailed review of communication strategies in modular robots is presented.

- **Chapter 3** details the hardware concept design of the Mo* modular robot platform. Requirements are set out for the system, and design decisions are detailed that work towards a design that can be implemented to fulfil these.

- **Chapter 4** discusses the implementation of the Mo* modular robot platform. The design decisions described in chapter three are translated in to an implementable design. The details of this implementation are described.

- **Chapter 5** explores the simulation platform developed to provide an environment to run experiments and develop software for the Mo* modular robot platform. The architecture of the simulation is described.

- **Chapter 6** details experiments performed to assess the effectiveness of the implementation of the Mo* modular robot platform. The details of the experiments are set out and results are discussed.

- **Chapter 7** presents the conclusions of this thesis and suggests some ideas for future research. The conclusions include the key achievements of the system, as well as some limitations that have been observed. The future work section discusses some system specific aspects that could be researched further, and some ideas for applications of the Mo* platform.

Chapter 2

Literature Review

2.1 Chapter Overview

In this section the author investigates the development of the field of modular and reconfigurable robots from the inception of the concept through to the current state of the art in the field. Initially a holistic review of systems is performed, investigating key components of each system, their target applications and performance. Next, the communication systems used in these systems are investigated in detail.

2.2 History of Modular Robotics

Modular robotics was first introduced under the term 'Cellular robotics' in 1990 by Fukuda et al [2]. The goal was to create a system that could reconfigure itself to an optimal structure for different purposes and environments. This work opened the door for research into robotic systems that could be built from similar and dissimilar autonomous components that could communicate with one another to co-operate and complete tasks.

After the introduction of CEBOT, research into modular robotics increased. The value of a robotic system that could alter its morphology to adapt to certain environments and tasks was clear. The timeline shown in Figure 2.1 depicts the release of new modular robotic systems by the year they were published in. After the introduction of CEBOT, it took a couple of years before other researchers began publishing material on modular robotics too. Since then there has been an increasing uptake in modular robotic research to this day.

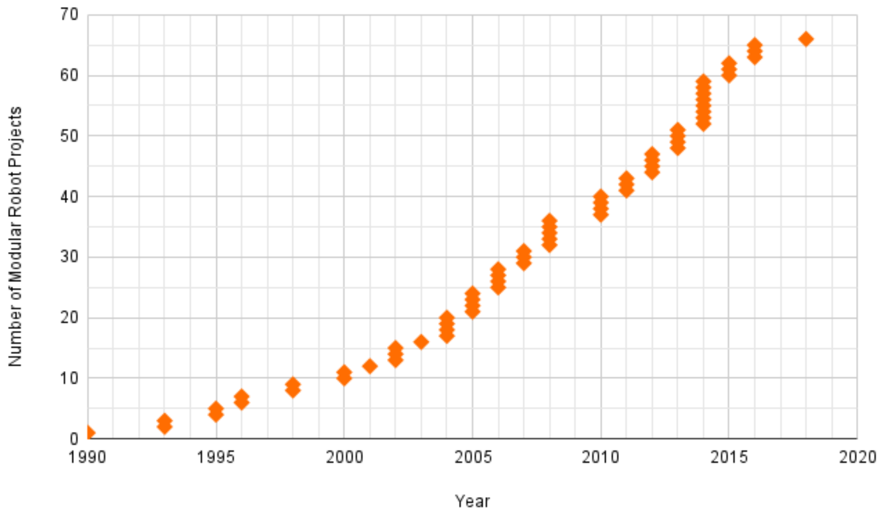


Figure 2.1: Modular robot system development over time

2.3 Existing Modular Robotic Systems

There have been many modular robotic systems produced since CEBOT, with efforts focused on varying applications and challenges from space exploration [3] [11], search and rescue [12] and others [9]. Initially, chain type systems were explored, where modules connect together in branching structures. As technologies developed lattice systems were explored, where modules can form two or three dimensional grid structures. In chain type systems it is easier to produce kinematic chains of modules for rolling treads or manipulators. The advantage of lattice systems is that they can re-configure more easily. It can be difficult to discretely classify the wide range of modular robot systems that have been developed. Ahmadzadeh et. al. attempted to create a catch-all classification system with the MITE framework in [9]. This classification system ended up with 120 different categories. Some platforms encompass a number of different categories. In this review the following categories will be used.

The first grouping of modular robotic systems is as follows.

- **Homogeneous** - These are systems where all modules are identical in both mechanical and electronic structure.
- **Heterogeneous** - These are systems where modules can take a variety of forms, connecting together through a standardised interface.

Two main categories have emerged referring to the assembly methodology;

- **Chain** In a chain type modular robot, modules are connected in linear and branching structures. This allowed systems of modular robots to form kinematic chains such as legs, arms and tracks. This category has been widely used for researchers developing self mobile homogeneous systems.

- **Lattice** Lattice type systems form structures in two or three dimensional grids. Each module can connect to multiple neighbouring modules. Lattice based systems can form more complex structures than chain based systems.

In the context of this thesis, it is useful to group modular robots by communication type, the following broad categories are used

- **Wireless** - Wireless standards and protocols include but are not limited to; Infrared, WiFi, Zigbee and Bluetooth.

- **Wired** - Wired protocols require an electrical connection to be present between modules, which increases the mechatronic complexity of the docking connector. Usually a system of spring loaded contacts are used to facilitate this.

  - **Bus Based** Bus based protocols use a communication bus shared between all nodes on the network. These include but are not limited to Controller Area Network (CAN), Inter-integrated circuit (I2C) and Serial Peripheral Interface (SPI)
  - **Point to Point** In point to point network topologies, nodes are connected together via isolated links. Some form of routing is required in these networks, either through dedicated routers or in an Ad-Hoc fashion where each node in the network also acts as a router to forward packets or reserve routes in a circuit switched scenario.

2.4 Survey Papers

Review papers compile, summarize, critique and synthesize information on a particular topic. There have been a number of review papers published on various aspects of modular robotics systems. These are summarized in Table 2.1.

An outline of the key points of each paper are provided in the following section then a comparative analysis of the papers is performed.

Year	Title	Authors	Citations
2003	Modular reconfigurable robots in space applications [6]	Mark Yim, Kimon Roufas David Duff Ying Shang, Craig Eldershaw, Sam Homans	356
2007	Modular self-reconfigurable robot systems [4]	Mark Yim, Wei-Min Shen, Behnam Salemi, Daniela Rus, Mark Moll, Hod Lipson, Eric Klavins, Gregory S. Chirikjian	1047
2008	Self-assembly at the macroscopic scale [13]	Roderich Groß, Marco Dorigo	177
2011	Modular and reconfigurable mobile robotics [7]	Paul Moubarak, Pinhas Ben-Tzvi	174
2015	Review of the modular self reconfigurable robotic systems [14]	Jacek Feczko, Michal Manka, Pawel Krol, Mariusz Giergiel, Tadeusz Uhl, Andrzej Pietrzyk	23
2016	Modular Robotic Systems: Characteristics and Applications [9]	Hossein Ahmadzadeh, Ellips Masehian, Masoud Asadpour	111
2017	Evolutionary Modular Self-Assembly and Self-Reconfigurable Robotics: Exhaustive Review [15]	Reem Alattas	1
2017	Current Trends in reconfigurable modular robots design [8]	Alberto Brunete, Avinash Ranganath, Segio Segovia, Javier Perez de Frutos, Miguel Hernando, Ernesto Gambao	115
2018	Toward growing robots: A historical evolution from cellular to plant-inspired robotics [16]	Emanuela Del Dottore, Ali Sadeghi, Alessio Mondini, Virgilio Mattoli, Barbara Mazzolai	52
2018	A review of Coupling Mechanism designs for modular reconfigurable robots [17]	Wael Saab, Peter Racioppo, Pinhas Ben-Tzvi	45
2021	Modularity for the future of space robotics: A Review [3]	Mark A. Post, Xiu-Tian Yan, Pierre Letier	22

Table 2.1: Review papers on the subject of modular robotics

Modular Reconfigurable Robots in Space Applications (Yim et al)

This paper from Mark Yim et al makes the case for using re-configurable modular robots for space applications. The authors state three characteristics of modular robots that are advantageous to space equipment; compactness and lightness, robustness, versatility and adaptability. The authors attribute modular robots with each of these characteristics. The paper discusses the advantages of having modular robots in space applications such as having the ability to perform a wider range of tasks with a given set of modular hardware than could be achieved with monolithic platforms.

Modular Self-Reconfigurable Robot Systems, Challenges and Opportunities for the Future (Yim et al)

Definition: "The field of modular self-reconfigurable robotic systems addresses the design, fabrication, motion planning, and control of autonomous kinematic machines with variable morphology"

Goal of Paper: Outline progress and identify key challenges and opportunities that lay ahead

The outline of the architecture of modular robotic systems is vague in this paper. Words such as "usually" and "often" are used to describe the frequency of occurance of certain components in a modular robotic system. The two main architectures, lattice and chain/tree, are described. Mobile architectures is stated as a third, independent category. Two reconfiguration categories are explored; deterministic and stochastic. Versatility, robustness and low cost are stated as the three key motivations for modular robots. The paper goes on to describe 'challenges for the future'. A detailed discussion is carried out on what the different challenges in creating effective modular robot platforms will be, and what opportunities having effective modular robot platforms would present.

Self-Assembly at the Macroscopic Scale (Groß et. al.)

This paper by Groß et al focuses on modular robots that have the ability to self from individual modules to connected structures. It defines self assembly as "a process by which preexisting components ("separate or distinct parts of a disordered structure") autonomously organize into patterns or structures without human intervention". The paper discusses the experience gained from designing 21 such systems, which encompass a wide range of components, including both passive mechanical parts and mobile robots. The authors provide a taxonomy for categorizing these systems and delve into the design principles and functions that govern their behavior. Ultimately, the paper outlines the major accomplishments in the field and suggests potential avenues for future research.

Modular and Reconfigurable mobile robotics (Moubarak et al)

This paper highlights the motivation and context for the research on robotic platforms with high autonomy and versatility. As demand for reliable robots capable of performing tasks and easing daily burdens increases, researchers have shifted their focus toward developing platforms that possess advanced capabilities and adaptability. These robots can function individually or collaboratively, and their modular structural morphology enables them to navigate and operate effectively in complex and unstructured real-world environments. The paper states that future research is needed in Docking and coupling, autonomy, locomotion patterns, modular manipulation and field testing. An analysis of the benefits and shortcomings of different modular robot architectures is performed.

Review of the modular self reconfigurable robotic systems (Feczko et al)

This paper begins by stating the current limitations of modular robotic systems for field use. It outlines the taxonomy of modular robot architectures as lattice, chain/tree and mobile architectures. Like the paper "Modular Self-Reconfigurable Robot Systems, Challenges and Opportunities for the Future" by Yim et al, it identifies deterministic and stochastic reconfiguration strategies. It identifies application areas as search and rescue, space, medical devices and "bucket of stuff", implying the field would like to get to the point where a generic library of modules could exist that can configure to complete a much wider variety of tasks than any monolithic system.

Modular robotic systems: Characteristics and Applications (Admadzadeh et al)

This paper is a well structured review of modular robot systems. It addresses the challenges of using Modular Robotic Systems (MRS) across various domains and proposes a comprehensive study of MRS characteristics and applications. The authors introduce a novel framework named MITE, which encompasses Module, Information, Task, and Environment perspectives. This framework is based on over 120 domain-specific features, offering a structured way to understand MRS properties and applications.

Evolutionary Modular Self-Assembly & Self Reconfigurable Robotics: Exhaustive Review (Alattas)

This paper provides a review of self-assembly, self-reconfiguration, self-repair and self-reproducability in the field of modular robots. This paper is focused on the use of evolutionary algorithms for completion of these tasks. The work includes a quantitative analysis of a selection of modular robot systems. A detailed description of these robot platforms is provided, with some analysis undertaken. Little discussion is given on the use and implementation of the use of evolutionary techniques.

Current trends in reconfigurable modular robots design (Brunete et al)

This review paper from Brunete et. al. provides a very detailed overview and analysis of various aspects of modular robotic systems. It outlines the various important attributes of modular robots and details the platforms that use each attribute. Locomotion strategies are reviewed in detail. Control strategies are also reviewed in this paper. There is a section on communication between modules that provides a good overview of communications methods in modular robotic systems. Power sharing between modules is also reviewed. This is a very detailed and complete review of the field of modular robots.

Towards Growing Robots: A Historical Evolution from Cellular to Plant-Inspired Robotics (Del Dottore et al)

This paper is not specifically on the subject of modular robots, but includes a good review of the field in the context of creating robotic systems inspired by cellular systems. The paper cites macro-scale modular robots as an important step in the path to creating microscopic systems of robot 'cells' that can assemble to mimic biological organisms.

A review of coupling mechanism designs for modular reconfigurable robots (Saab et al)

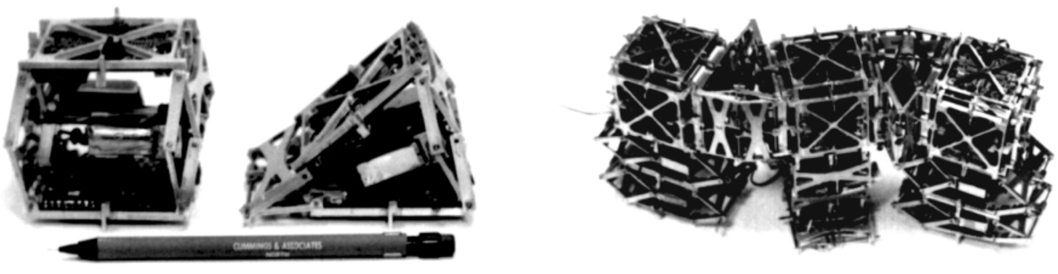
This paper by Saab et. al. provides a detailed review of coupling mechanisms, or docking connectors for connecting modular robots together to form larger systems. The paper lists and describes the design attributes that can be assigned to modular robot docking mechanisms. It reviews the different classifications of docking connectors by group. The paper finishes with a tabulated summary of the modular robot connection mechanisms reviewed. The paper is not able to draw a conclusion as to the best type or implementation of a docking connector, citing advantages and disadvantages for each.

2.5 Modular Robot Platforms

CEBOT

CEBOT was introduced by Toshio Fukuda (Nagoya University) and Yoshio Kawauchi (Science University of Tokyo) in 1990. The aim of this project was to create a robotic system that could be reconfigurable for a variety of manipulation tasks. CEBOT used COMBUS for docked communications. The combus protocol used a 14 pin connector. The connector is of gendered type, so modules can only connect in one direction. Out of the 14 pins on the combus connector, 12 are used for communications. The communication is of serial type. The 12 bits are divided into 3 groups of 4 bits. H-Digit, M-Digit and L-Digit. These are for data, sender cell address and receiver cell address respectively. This means that the system has a theoretical maximum of 16 modules, as that is the amount of address space. In experimentation, the authors were able to achieve XX bits/second. In an un-docked state, IR communication is achieved with 8 parallel bits. The authors of CEBOT were aware that there is a real need in modular robotics to strictly define an efficient method of storing information. They stipulated that the more information that is in the system, the less communication will be required. This statement only holds true if every module in the system holds that knowledge, which may be less efficient than sharing knowledge as and when agents in the system need the knowledge.

Polypod

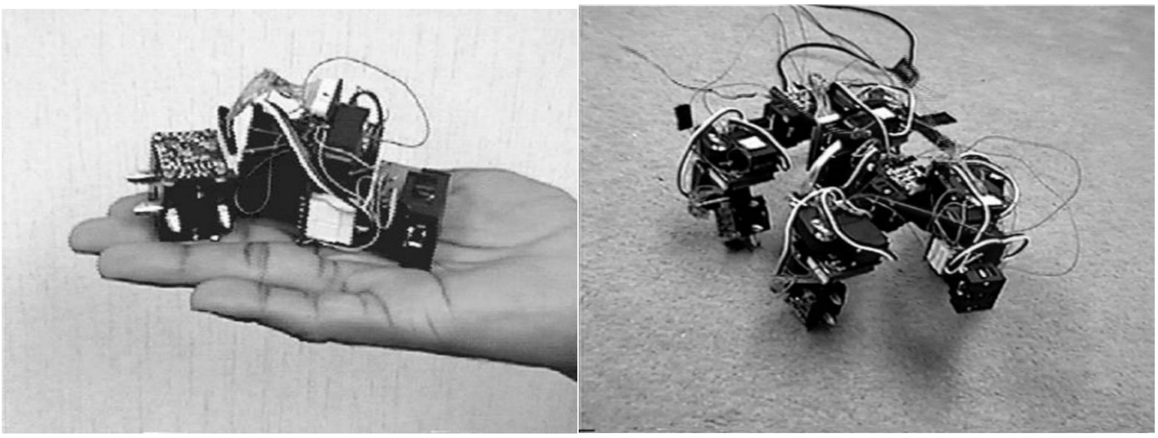


(a) Polypod Cell (b) Polypod in Caterpillar Configuration

Figure 2.2: Polypod Modular Robot [18]

Mark Yim developed the polypod modular robot at Standford University in 1994 as part of their PhD dissertation. Polypod was created for the purposes of testing locomotion gaits. Polypod uses SPI over RS485 for inter module communication. Imposing SPI onto an RS485 channel aids in increasing noise immunity. Infrared photodiodes are used for undocked communication. The processor used in polypod modules is the motorola MC68HC11E2, which is an 8-bit processor running at 2MHz. Each module face has 2 IR photodiodes, 4 IR LEDs, 4 male electrical connectors and 4 female electrical connectors. This gives four times symmetry around the x axis of the face.

CONRO



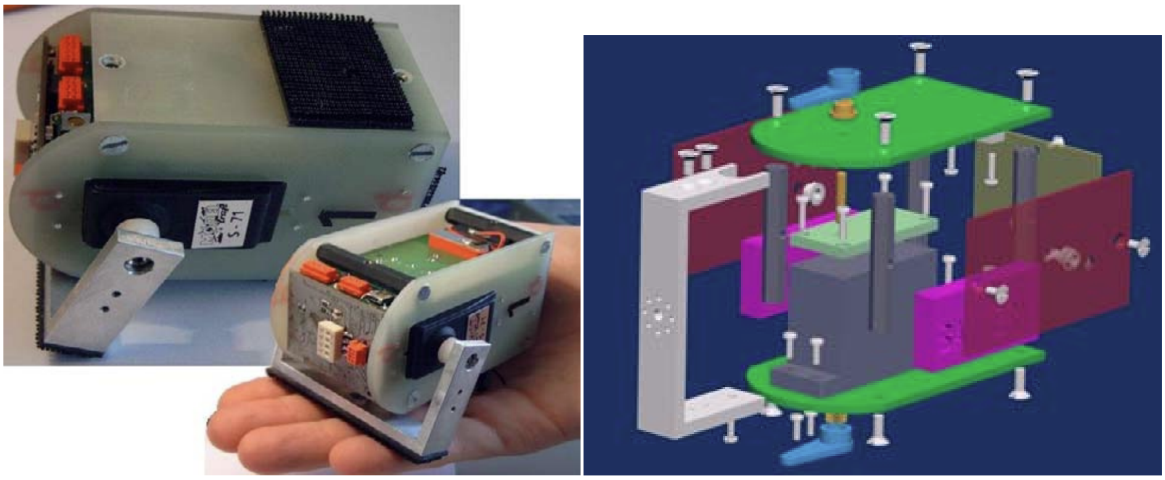
(a) CONRO Module (b) CONRO configured as a Quadruped

Figure 2.3: CONRO Modular Robot[19]

Castano et al Developed the Conro modular robot platform in 2002 at the University of Southern California. The Conro platform aimed to develop modular robots that can reconfigure into different shapes such as snakes and hexapods. Each module has it's own processor, power supply, communcations, sensors and actuators. Conro is primarily a homogeneous system, although some modules with extra sensors were created. Each module is comprised of three segments, giving two independent degrees of freedom. Conro uses a pin and latch connection mechanism. Infrared transceivers are used for inter module communication.

YaMoR

In the field of modular robotics, the only system that utilizes an FPGA for processing is YaMoR (Yet another Modular Robot) developed by Daniel Marbach et al at the Swiss Federal Institute of Technology. YaMoR was developed to test some theories of using central pattern generators for locomotion control. Communication between YaMoR modules is achieved using Bluetooth for both docked and un-docked states. There is a dedicated PCB within the



(a) YaMoR Module (b) Exploded CAD view of Yamor Module

Figure 2.4: YaMoR Modular Robot[20]

module consisting of an ARM SoC for controlling Bluetooth communications. This SoC communicates with the FPGA controller board using UART Communications. This limits the communication throughput in the system to that of the UART Channel, as Bluetooth has a higher throughput than UART. The FPGA on the controller board is a Xilinx Spartan 3 XC3S400 running at 50MHz. This particular FPGA consists of 400,000 logic gates. The logic design for YaMoR is of particular interest. There consists of a mixture of fixed logic as well as reconfigurable logic. The fixed logic regions contain a microblaze RISC-V soft processor as well as IO handling circuits. The purpose of the microblaze processor is to control the loading and unloading of partial bitstreams into the dynamically reconfigurable regions of the device. Partial bitstreams can be shared throughout the whole robotic system. The YaMoR modules do not have a dynamically reconfigurable morphology. Alterations to the morphology of the system is done manually. The modules attach with hook-and-loop style velcro fasteners. This is a genderless connection. The modules do not share any electrical connections, each module has its own power source.

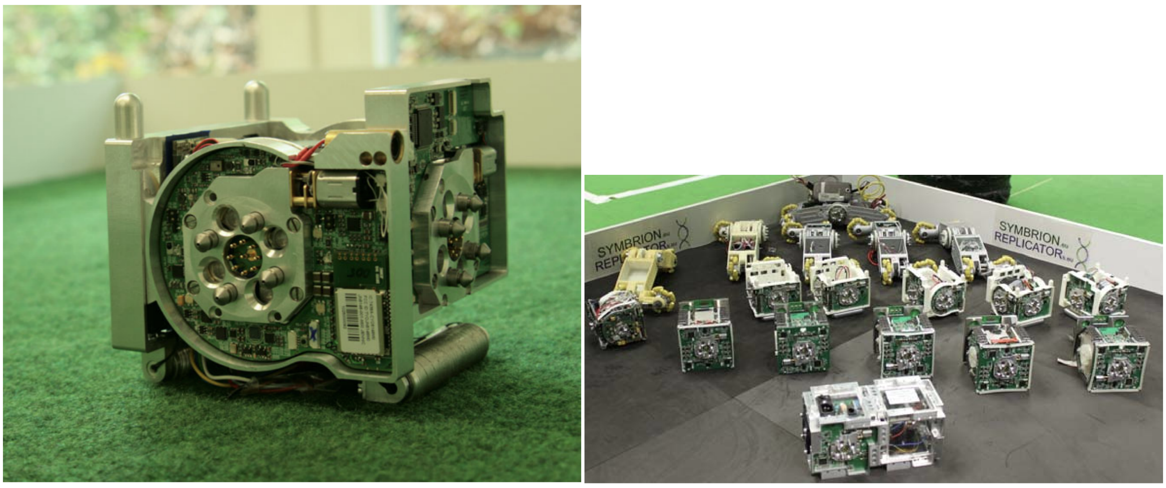
Superbot

Superbot was developed by Salemi et al at the University of Southern California. This is one of the most robust modular robot platforms that have been developed to date. Superbot is a homogeneous platform that was developed to produce flexible, powerful and sturdy modules that could efficiently perform tasks in uncontrolled environments without requiring close attention. The focus of the Superbot system was on locomotion strategies. Each module has three degrees of freedom. It uses infrared transceivers for inter module communication.

(a) Single Superbot Module (b) Superbot in humanoid configuration

Figure 2.5: Superbot Modular Robot [21]

CoSMO



(a) CoSMO Module (b) Exploded CAD view of Yamor Module

Figure 2.6: CoSMO Modular Robot [22]

Liedke et al. at the Karlsruhe Institute of Technology in Germany developed the CoSMO (Collective Self-reconfigurable Modular Organism) as part of the SYMBRION project. This modular system is the only system, to the author's knowledge, that uses ethernet for communications between modules. This gives the capability for throughput of 100MBits/s. This modular robot project aimed to push the boundaries of processing power and communication in modular robotic systems. This CoSMO module is one of three heterogeneous module types designed and built for the symbricator modular robotic system. For processing, the module contains a Blackfin analog digital signal processor, five MSP430 MCUs for peripheral sensor preprocessing, and an arm coretex based processor for 3d locomotion calculations. The Blackfin processor is clocked at 550 MHz. The use of MSP430 MCUs for sensor preprocessing relieves the main processor of recurring processes that could reduce completion time of other processes. Each of these processing devices communicate via an intra module SPI bus. Each of the four connector faces in the module has an ethernet port. These ethernet interfaces are connected to an off the shelf five port ethernet switch IC with the final port being connected to the Blackfin processor. There is also the capability to perform wireless Zigbee based communication between modules. Should the SPI and/or ethernet communication interfaces fail, or a processor becomes locked up, the module has a fail safe strategy using an I2C communication bus to move to a 'safe mode', The modules in the system can share power as required, with one module being able to recharge other module's batteries upon request. For the software side, the Blackfin processor runs uClinux for an operating system. processes in the application communicate via SOAP calls. A publish and subscribe mechanism is used for accessing sensor data. The developers of CoSMO have created an API to make creating application code for the system easier for future developers.

Kairo 3

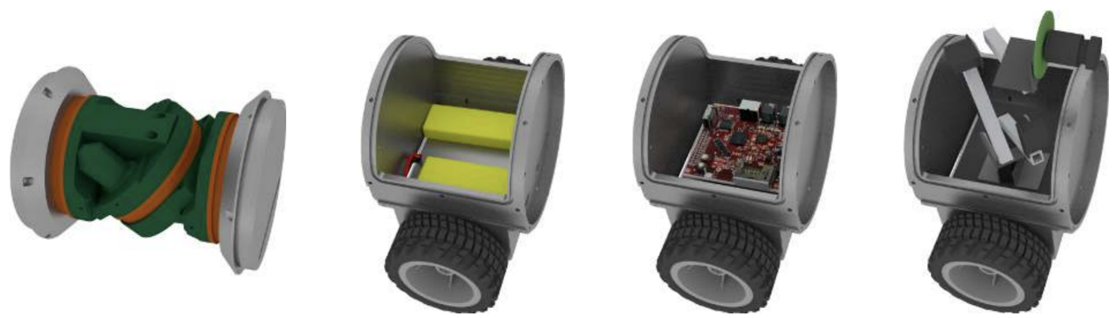


Figure 2.7: Kairo 3 heterogeneous modules [23]

Kairo 3 was developed at the department of interactive diagnosis and service systems at the FZI Research center for information technology, Karlsruhe, Germany. The first Kairo 3 paper was released in 2014. It is based on work from previous KAIRO (1997), MAKRO (1999), MAKRO PLUS (2004) and KAIRO-II (2008) models developed at the same institute. Its primary aim is to be a platform for search and rescue field missions.

Kairo 3 is a snake like reconfigurable inspection robot. It is a chain type structure inspired by an inchworm and snake. It is heterogeneous as it has two different types of modules in the system. These are drive modules and joint modules. Each joint module has 3 DOF.

Harmonic drive gears are used with a gear ratio of R=160. Max torque of 4.8 Nm per axis. Optical encoders used for joint angle feedback. Typical config of KAIRO 3 consists of six drive modules interleaved with 5 joint modules. Drive modules weigh 4.5 Kg and are 146 mm long. Joint modules are 4 Kg and 184 mm long. For this typical config overall length is 1.8 m and mass is 47 Kg including batteries. Can climb 55 cm steps, pass through 25 cm diameter pipes.

MCA2 (Modular controller architecture) is a novel system architecture implementation used in Kairo. A two layer architecture is used, one for per module control and one for system control. A UCom (Universal controller modules) board is used in each module. CAN bus is used to communicate between modules. System control is done on a beagelboard in a drive module. Adaptive control software employed for automatic locomotion adaption to varying terrains. Only manually reconfigurable. Catalog of modules developed including Joint, battery, embedded PC and manipulator modules.

Summary

The modular robot platforms described in this section have each represented a significant advancement in the field. While each of these advancements are novel and useful for the field, the platforms that have been developed as part of those projects have been tailored to showcase the particular advancement of interest. This has left the field of modular robots with a wide variety of modular robot platforms that are each good at different things. The aim of modular robots is to make robotic platforms that are more adaptable and resilient than monolithic platforms, and the majority of the platforms identified in this review do not fulfil that criteria. The notable exception to this is the CoSMo platform, which strove to create a truely heterogeneous modular robot platform based on a standardised interface. This system, however, has become complex to design and manufacture around, this work aims to take those principles and create a simpler implementation to fit to that set of criteria.

2.6 Docking Connectors in Modular Robots

2.6.1 Design of Docking Connectors

The primary function of a docking connector in modular robot applications is to mechanically attach two modules together through a standardised interface. This holds for both homogeneous and heterogeneous systems. A secondary function that is desirable is to provide electrical connections between modules to allow power and data to be shared throughout the system. Without this functionality, every module in the system is required to contain a battery and a means of wirelessly communicating. Another important feature of docking connectors is whether they are gendered or genderless. With gendered docking connectors modules must connect a male connector to a corresponding female connector, limiting the flexibility of the system. Gendered connectors are easier to design and are usually used in branching chain architectures. With genderless connectors, any module face can connect to any other module face. This greatly increases the flexibility of the system and the number of connection permutations. Rotational symmetry also plays an important role in docking connectors. Ideally docking connectors will be rotationally symmetric for the number of edges of the connector body outline, so for a square connector, this would be four times rotationally symmetric. Figure 2.8 shows the spread of docking methods used across the reviewed platforms.

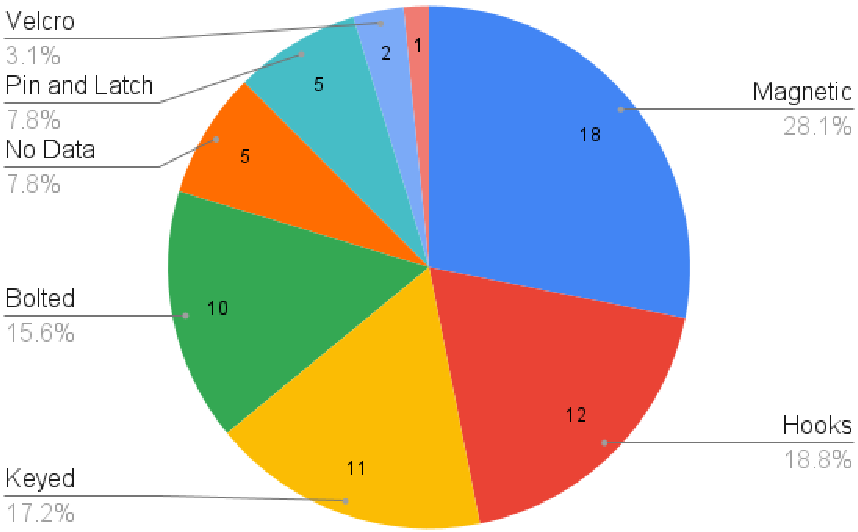


Figure 2.8: Share of Module Connection Mechanisms used in modular robots

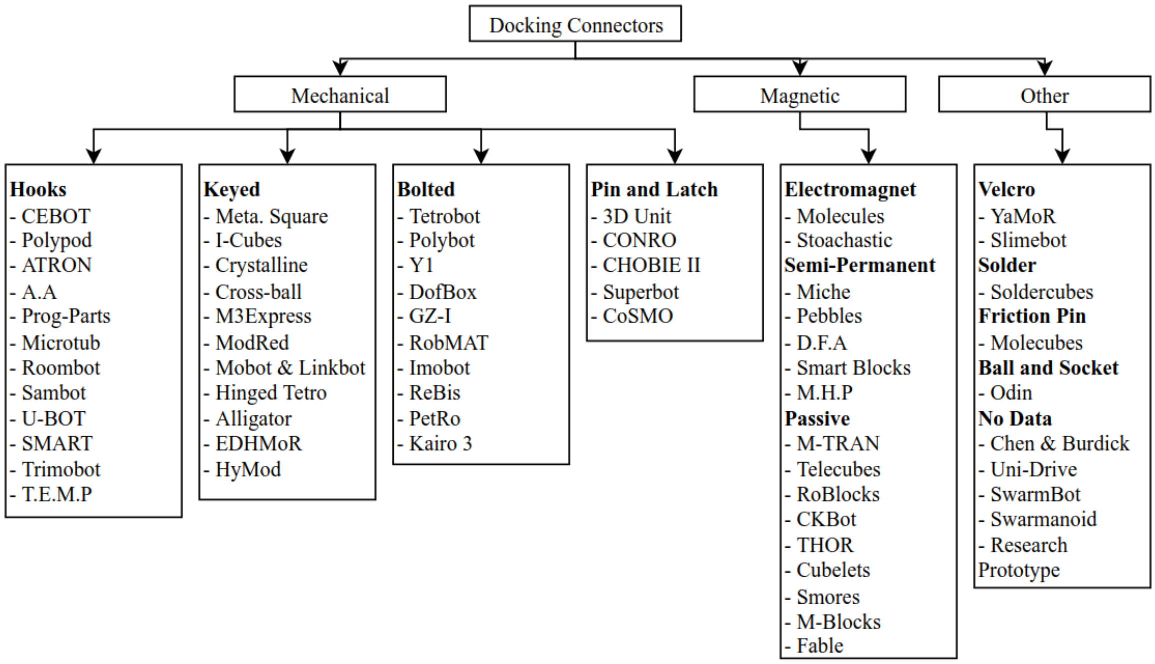


Figure 2.9: Classes of docking Mechanisms

Figure 2.9 depicts the different classes of docking connectors for the modular robots reviewed. These are described in more detail below. The data used to create these classes is tabulated in Table 2.2.

Magnetic

Magnetic docking connectors use either permanent, semi-permanent or electro-permanent magnets to provide a holding force between two modules. Magnetic connectors are usually mechanically static, making design and construction easier than other methods. Fracturm [24] was the first modular robot to use a magnetic docking mechanism. It consists of three planar triangular sections that are arranged in an offset pattern. The two outer planes contain permanent magnets, and the middle layer contains electromagnets used for actuating connection between modules. M-Blocks [25] is another notable platform to use magnetic docking mechanisms. M-blocks is a momentum driven modular robot that uses an flywheel to move modules relative to each other. Each module contains 3 permanent magnets in each edge of the cubic structure. The flywheel is rapidly accelerated to overcome the magnetic attraction force between one side of a face, and so a module can rotate onto an adjacent face of a neighboring module.

Hooks

Hooks are a commonly used method of connecting modules together in a modular robotic system. Generally hooked based systems are gendered, as it is mechanically complex to implement a genderless hook based system. Hooked based systems are generally actuated by a DC motor on the male side, with the hooks 'grabbing' a post or latch on the female side. The first modular robot developed, CEBOT [26][2][27], used a hook mechanism for connection. A male connector is actuated by a DC motor with a worm gear coupling to two hooks that latch to corresponding latching posts on a female connector.

Keyed

Keyed based connection mechanisms use an especially designed shape on the male side that is inserted into a corresponding cavity on the female side. This shape is usually then rotated to lock it into the female side. Keyed systems are most often gendered based designs. A keyed based coupling mechanism is easy to design and construct with FDM printing techniques. A downside of these systems is that they are bulky, and do not cope well with misalignment between modules. Crystalline [28][29] used a gendered key based mechanism. it is a two-dimensional lattice modular robot. Each module contains four docking faces, two male, with actuated key, and two female. Once inserted into a corresponding female connector, the key rotates 90 degrees to lock the two modules together.

Pin and Latch

Pin and latch mechanisms use keyed pins to insert into corresponding holes on an adjacent module. A spring loaded or electro-mechanically actuated latch is then used to lock the two modules together. Pins can have a conical tip to allow for more alignment tolerance between two connectors. Conro[19][30][31] uses a pin and latch mechanism that is gendered. An SMA wire is used to actuate the latch mechanism. Conro connectors only have two rotational degrees of symmetry.

Bolted

Bolted mechanisms include any modular robot platform that cannot actuate it's own docking connector and requires human intervention to bolt two modules together. Many early modular robot platforms used this method as the research was focused on designing locomotion patterns for homogeneous systems.

Summary

This section has outlined an in-depth review in to mechanical connection strategies used in modular robot platforms. Docking connectors have been categorised on two different levels; the first level is split in to mechanical, magnetic and other types of docking connectors. Within the mechanical connector category there are four sub-categories; hooks, keyed, bolted and 'pin and latch'. Within the magnetic category there are three sub-categories; electromagned, semi-permanent and passive. The other category includes any docking mechanisms that do not fit in to one of the other categories. The hooked type of docking mechanism is the most widely used. This is likely because you can generate a significant holding force from hooked mechanisms against both translation and rotational forces. The problem with this type of mechanism is that is can be mechanically complex to implement, with many small parts involved. Overall there is no clear 'best practice' for docking connector design, as there is a relatively even spread of approaches used across the categories presented.

Table 2.2: Modular Robot Docking Mechanisms

Name	Year	Interface	Gender	Symmetry	Actuation
CEBOT	1990	Hooks	-	-	-
Metaporphising Robot	1993	Magnetic	-	-	-
Polypod	1993	Hooks	Genderless	2	SMA
Fracta	1995	Magnetic	Gendered	1	DC Motor
Chen & Burdick	1995	-	-	-	-
Tetrobot	1996	Bolted	Genderless	1	Manual
Meta. Square	1996	Keyed	Gendered	1	DC Motor
Molecules	1998	Magnetic	Gendered	1	Electromagnet
3D UNIT	1998	Pin and Latch	Gendered	1	DC Motor
I-Cubes	2000	Keyed	Gendered	4	DC Motor
Polybot	2000	Bolted	Genderless	4	Manual
Crytalline	2001	Keyed	Gendered	1	DC Motor
CONRO	2002	Pin and Latch			
M-TRAN	2002	Magnetic	Genderless	2	Passive
Telecubes	2002	Magnetic	Genderless	4	Passive
Uni-Drive	2003				
ATRON	2004	Hooks	Gendered	1	DC Motor
Automatic Assembly	2004	Hooks	Gendered	4	DC Motor
CHOBIE II	2004	Pin and Latch	Gendered	1	DC Motor
Stoachastic	2004	Magnetic	Gendered	1	Electromagnet
Porg-Parts	2005	Hooks	-	-	-
YaMoR	2005	Velcro	Genderless	1	Manual
Swarmbot	2005	-	-	-	-
MicroTub	2005	Hooks	Gendered	-	-
Superbot	2006	Pin and Latch	Gendered	4	Manual
Y1	2006	Bolted	Genderless	4	Manual
RoBlocks	2006	Magnetic	Genderless	4	Passive
Slimebot	2006	Velcro	Genderless	-	Manual
Molecubes	2007	Friction Pin	Genderless	4	Manual
CKBot	2007	Magnetic	Genderless	4	Passive
DofBox	2007	Bolted	Gendered	4	Manual
Miche	2008	Magnetic	Genderless	4	S.P Magnet
Roombot	2008	Hooks	Genderless	4	DC Motor
GZ-I	2008	Bolted	Genderless	4	Manual
Odin	2008	Ball and Socket	Gendered	6	Manual
RobMAT	2008	Bolted	Genderless	4	Manual
Sambot	2010	Hooks	Genderless	4	DC Motor
Imobot	2010	Bolted	Genderless	4	Manual
Pebbles	2010	Magnetic	Genderless	4	S.P Magnet
THOR	2010	Magnetic	Genderless	8	Passive
Cross-ball	2011	Keyed	Genderless	2	DC Motor
D.F.A	2011	Magnetic	Genderless	1	S.P Magnet
U-BOT	2011	Hooks	Gendered	4	DC Motor
Cublets	2012	Magnetic	Genderless	4	Passive
M3Express	2012	Keyed	Gendered	4	Servo
SMART	2012	Hooks	Genderless	4	DC Motor
Smores	2012	Magnetic	Genderless	2	Passive
CoSMO	2013	Pin and Latch	Genderless	4	DC Motor
M-Blocks	2013	Magnetic	Genderless	4	Passive
Smart Blocks	2013	Magnetic	Genderless	1	S.P Magnet
Swarmanoid	2013	-	-	-	-
Trimobot	2014	Hooks	Gendered	1	DC Motor
ReBis	2014	Bolted	Gendered	2	Manual
PetRo	2014	Bolted	Gendered	1	Manual
ModRed	2014	Keyed	Genderless	4	Stepper Motor
Mobot and Linkbot	2014	Keyed	Gendered	1	Manual
Kairo 3	2014	Bolted	Gendered	1	Manual
Hinged Tetro	2014	Keyed	Gendered		Servo
T.E.M.P	2014	Hooks	Gendered	1	DC Motor
Research Prototype	2015		-	-	-
Alligator	2015	Keyed	Gendered	1	SMA
EDHMoR	2015	Keyed	Genderless	4	DC Motor
HyMod	2016	Keyed	Genderless	4	DC Motor
Soldercubes	2016	Solder	Genderless	4	Resistive Heating
Fable	2016	Magnetic	Genderless	4	Passive
M.H.P	2018	Magnetic	Gendered	1	S.P Magnet

2.7 Communication Mediums in Modular Robotics

Throughout the 30 year history of modular robotics research, authors have chosen to adopt many different communication standards for their platforms. As the underlying embedded technologies have evolved, so have the physical layers and protocols used. Communication methods can be split in to three fundamental groups of communication medium, Wired, Radio Frequency and Magnetic. These are summarised below.

Wired

Wired communications involved any communication standard that uses electrical conductors making contact to transmit data between modules. This can be achieved with spring loaded pogo pins or male and female connectors.

Radio Frequency

Radio frequency wireless communications involve any communication standard that uses the radio frequency spectrum with free air as the medium to transmit information. Wireless communication makes module connector design easier, as there is no need to have electrical contacts for communication on the face of the connector. Depending on the multiplexing used by the data link protocol, radio spectrum can fill fast meaning the communication system mightnot be scalable to a large number of modules.

Magnetic

Magnetic communications involve using electromagnets to impose a magnetic field then converting that to an electrical current with another electromagnet.

Summary

Wired communication is the most used medium in modular robotics. platforms can implement an electrical connection between modules in a variety of ways, and as little as two conductors are needed for serial protocols to operate. Wireless communication has been used in a number of platforms. For each of these mediums, there are protocols that can be used to define how the medium is used, and how data is send across it. The protocols found in the literature for modular robotic systems are described in the following section.

2.8 Communication Protocols in Modular Robotics

Figure 2.10 shows the usage of different communication standards across all the modular robotic systems reviewed. The data used for this is tabulated in 2.3. Thirteen individual protocols were found to be used. It can be seen that infrared (IR) is the most widely used, with nearly a quarter of all modular robots using the technology. IR is a wireless protocol which is easy to implement in embedded systems, this is likely why it is the choice for many system designers. UART is the second most popular choice. This is one of the oldest and most ubiquitously adopted serial protocols in embedded systems. Nearly all microcontrollers produced over the last 30 years contain a UART peripheral. Each of the protocols listed are described in detail below.

The stacked area chart in Figure 2.11 depicts the usage of communication protocols in modular robots over time. The popularity of each protocol can be seen. UART was the first method used to communicate. Infrared quickly gained popularity after it was first used by Polypod in 1993 and remains the most used. As wireless technologies developed, their uptake in modular robotics followed. Zigbee, WiFi and Bluetooth are popular standards at present. CAN quickly showed popularity as a wired protocol and remains a widely used choice.

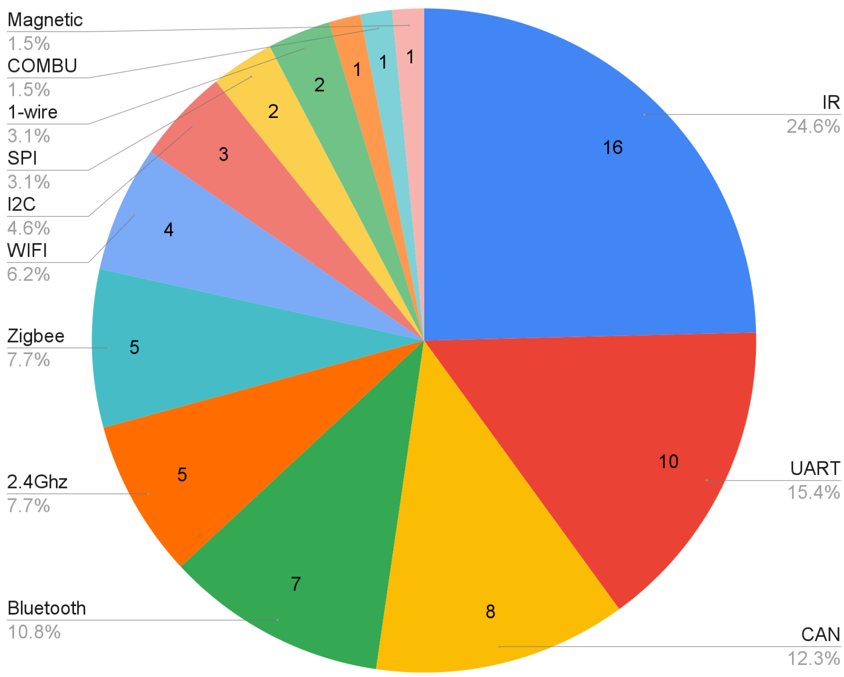


Figure 2.10: Share of communication protocols used in modular robots

Table 2.3: Communication in Modular Robots

Name	Year	Inter-Module A	Inter-Module B	Intra-Module
CEBOT [2]	1990	COMBUS	IR	-
Metaporphising Robot [32]	1993		-	
Polypod [18]	1993	IR	SPI	-
Fracta [24]	1995	IR	-	-
Chen & Burdick[33]	1995	-	-	-
Tetrobot [34]	1996	-	-	-
Meta. Square [35]	1996	-	-	-
Molecules [36]	1998	UART	-	-
3D UNIT [37]	1998	-	-	-
I-Cubes [38]	2000	UART	-	-
Polybot [39]	2000	CAN	IR	
Crytalline [29]	2001	IR	-	-
CONRO [30]	2002	IR	-	-
M-TRAN [40]	2002	UART	LonWorks	UART
Telecubes [41]	2002	IR	-	-
Uni-Drive [42]	2003	-	-	-
ATRON [43]	2004	IR	-	I2C
Automatic Assembly [44]	2004	-	-	-
CHOBIE II [45]	2004	IR	-	-
Stoachastic [46]	2004	UART	-	-
Porg-Parts	2005	?	?	?
YaMoR [47]	2005	Bluetooth	-	UART
Swarmbot [48]	2005	WiFi	-	I2C
MicroTub [49]	2005	I2C	-	-
Superbot [21]	2006	IR	SPI	I2C
Y1	2006	?	?	?
RoBlocks [50]	2006	?	?	?
Slimebot [51]	2006	IR	-	-
Molecubes [52]	2007	UART	-	UART
CKBot [4]	2007	CAN	IR	-
DofBox [53]	2007	I2C	-	I2C
Miche [54]	2008	IR	-	I2C
Roombot [55]	2008	?	?	?
GZ-I [56]	2008	I2C	-	-
Odin [57]	2008	UART	-	SPI
RobMAT [58]	2008	CAN	Bluetooth	-
Sambot [59]	2010	CAN	Zigbee	I2C
Imobot [60]	2010	WIFI	Bluetooth	?
Pebbles [61]	2010	Magnetic	-	-
THOR [62]	2010	UART	-	-
Cross-ball [63]	2011	?	?	?
Distributed Flight Array [64]	2011	UART	2.4Ghz Wireless	I2C
U-BOT [65]	2011	2.4Ghz Wireless	-	UART
Cublets	2012	?	?	?
M3Express [66]	2012	Bluetooth	-	?
SMART [67]	2012	CAN	Bluetooth	?
Smores [68]	2012	WIFI	-	I2C
CoSMO [22]	2013	Ethernet	?	SPI
M-Blocks [25]	2013	Zigbee	IR	I2C
Smart Blocks [69]	2013	-	-	-
Swarmanoid [70]	2013	IR	2.4Ghz Wireless	?
Trimobot [71]	2014	Zigbee	-	SPI
ReBis [72]	2014	UART	-	-
PetRo [73]	2014	IR	-	-
ModRed [74]	2014	Zigbee	-	I2C
Mobot and Linkbot [75]	2014	Zigbee	-	-
Kairo 3 [76]	2014	CAN	1-wire	-
Hinged Tetro [77]	2014	UART	-	-
T.E.M.P [78]	2014	WIFI	-	UART
Research Prototype	2015		-	
Alligator [79]	2015	Bluetooth	-	-
EDHMoR [80]	2015	CAN	2.4Ghz Wireless	-
HyMod [12]	2016	CAN	Bluetooth	I2C
Soldercubes [81]	2016	1-wire	-	1-wire
Fable [82]	2016	2.4Ghz Wireless	-	SPI
Modular Hydraulic Propulsion [83]	2018	-	-	-

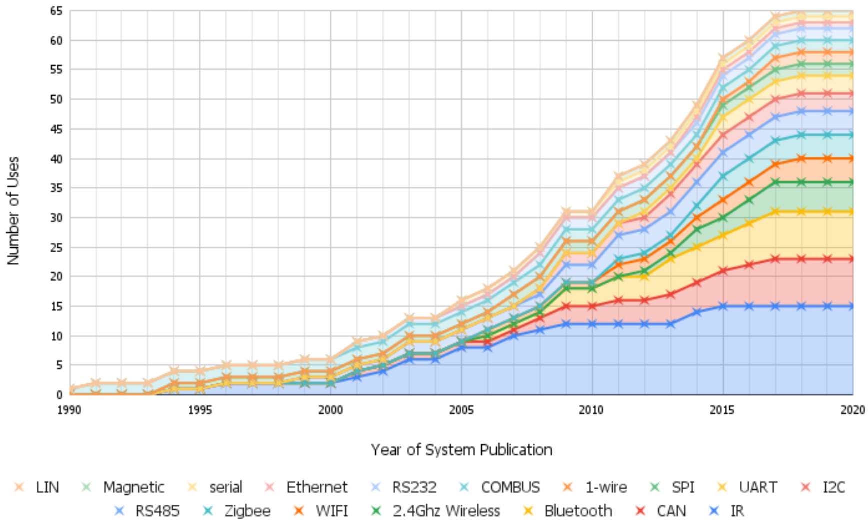


Figure 2.11: Protocol usage in modular robots over time

COMBUS

COMBUS was the communication protocol used in the first ever modular robotics platform. COMBUS is an 8 bit parallel data interface. The pin assignments are given in Table 2.4.The three bus function bits are mapped as shown in Table 2.5.

Table 2.4: COMBUS pin descriptions.

Pin-No	Name	Description
1	Bit 7	Data valid, active low
8	Bit 6	Bus function 2
2	Bit 5	Bus function 1
9	Bit 4	Bus function 0
3	Bit 3	Data bit 3
10	Bit 2	Data bit 2
4	Bit 1	Data bit 1
11	Bit 0	Data bit 0

Table 2.5: COMBUS Bus function bits.

Bit			Bus Function,
2	1	0	Data Word Type
0	0	0	Data low digit
0	0	1	Data high digit
0	1	0	Address low digit
0	1	1	Address high digit
1	0	0	Control low digit
1	0	1	Control high digit
1	1	0	Spare
1	1	1	Bus reset

All COMBUS communication sequences utilize a three step attention/data transfer/end of attention cycle. In CEBOT, a full communication sequence is cited to take 140 ms.

IR

Infrared communication involved using an infrared light LED to send data over free air and a photodiode to detect the magnitude of these light pulses to receive data. Users can quite readily implement their own software protocol for use with the IR physical layer, or choose to use a standard protocol. IrDA (Infrared Data Association) is a widely used communication protocol with integrated circuit manufacturers making IrDA interface ICs to ease integration into embedded designs. IrDA speeds range from 2.4 Kbit/s through to 1Gbit/s for the latest standard, although the implementation for speeds of 1Gbit/s is impractical for modular robotic systems. Infrared communications must have a clear optical line of sight between receiver and transmitter. Infrared communication was first used by CEBOT[2], and has subsequently been used by Polypod, Fracta, Polybot, Crystalline, CONRO, Telecubes, ATRON, CHOBIE II, Superbot, Slimebot, CKBot, DofBox, Miche, M-Blocks, Swarmanoid, PetRo and OmniPiTent.

CAN

CAN (Controller Area Network) (https://www.can-cia.org/can-knowledge/can/can-history/) is a serial bus-based communication protocol first introduced in 1986 for the automotive sector. Its aims were to provide a method of serial communication for devices in passenger cars without the need for much conductor mass. At the time electronic systems in automobiles were increasing in complexity, and wiring harnesses were getting more and more complex. CAN operates over two conductors, a high line and a low line. The maximum bit rate of CAN is 1 Mbit/s. A CAN bus must be terminated at each end with 120 ohm resistors. A common ground must be shared between nodes. CAN uses NRZ coding at the physical layer. CAN can operate in a 5V version or a 3.3 volt version. CAN was first used by Polybot in 2000, and has subsequently been used by CKBot, RobMAT, Sambot, S.M.A.R.T, Kairo 3, EDHMoR and HyMod.

Bluetooth

Bluetooth(r) is a wireless communication protocol that uses radio frequency bands from 2.402 GHz to 2.480 GHz as its medium. It was first developed in 1989 by Ericsson Mobile to use for wireless headsets. since then it has become one of the most popular standards in the world for personal area networks attached to mobile phones. Use cases include streaming audio to Bluetooth enabled speakers. Because it has mainly been developed for the consumer mobile market, Bluetooth operates at very low power consumption. A quasi optical wireless path must be visible between Bluetooth transceivers. Most Bluetooth transceivers have a range of up to 10 meters. The latest range of Bluetooth devices have transfer rates of up to 25 Megabits per second. nominal transfer rates are in the order of 3 Mbits/s. The Bluetooth standard is maintained by the IEEE under 802.15.1. Bluetooth was first used by YaMoR in 2005 and has subsequently been utilized in RobMAT, Imobot, S.M.A.R.T, M3 Express, Alligator and HyMod.

WiFi

WiFi is a wireless communication standard that uses the 2.4GHz and 5GHz radio frequency bands. It was first introduced in 1999 for use as a local area network between personal computers. Wifi modules can be purchased for easy integration into embedded systems. The most widely, and almost ubiquitously, used of these modules the ESP8266 which can operate at theoretical speeds of up to 600 Mbits/s, although nominal speeds are 3-54 Mbits/s. As the number of nodes connected and streaming data through the network increases, the data transfer rate decreases. WiFi was first used by swarmbot in 2005, and has since been used in imobot, smores and T.E.M.P.

Zigbee

Zigbee is a wireless communication standard that operates in 2.4 GHz to 2.4835 GHz radio frequency bands. It was first standardised in 2003 for use in low power, low datarate devices. It has a defined data rate of 250 Kbit/s and an operating range of 10 to 20 meters. There is a wide variety of Zigbee modules available for easy integration of the protocol into embedded systems. Zigbee is based on and maintained by the IEEE 802.15.4 standard. Zigbee was first used in modular robotics by Sambot in 2005, and has subsequently been used by M-Blocks, transmote, Trimobot, modRed and Mobot and linkbot.

I2C

I2C (IIC, Inter Integrated Circuit) is a wired communication protocol. It was first introduced in 1982 by phillips semiconductor. It is a synchronous multi-master serial bus based protocol. It is very widely used for connecting low speed sensor peripherals to microcontrollers. I2C uses two electrical lines, one for data and one for clock. Pullup resistors are required on the bus. I2C has up to a 10 bit address space meaning that theoretically 1024 devices could be connected to the bus. This would be impractical though as there would be a lot of contention on the bus and thus operate very slowly. Data rates can be up to 5 MBits/s on an I2C bus although nominal speeds are more commonly around 1 MBit/s. A common ground must be shared between nodes. Most microcontrollers produced at presemt have at least one

I2C device, meaning it is very easy to implement in an embedded system. I2C is not used widely for inter module communication in modular robotics due to its slow speed and lack of scalability. It was first used by Microtub in 2005, then subsequently Dofbox and GZi. I2C is commonly used for intra module communication, where a peripheral preprocessing microcontroller needs to connect to a CPU or similar architecture. I2C was first used for intra module communication by ATRON in 2004, then Swarmbot, Superbot, DofBox, Miche, Sambot, Distributed Flight Array, Smores, M-Blocks, ModRED and Hymod after that.

SPI

SPI (Serial Peripheral Interface) is a synchronous serial communication standard first developed by Motorola in 1985. SPI is a full duplex serial bus. Three seperate conductors are needed for a minimum implementation of SPI. A common ground must be shared between nodes. For each slave connected to the bus, a slave select line is required for the master to know which slave it is communicating with. SPI datarates can go up to and over 1 Mbit/s. SPI was designed primarily for microcontroller to microcontroller communication. There can be as may devices as there are slave select lines on an SPI network. SPI, unlike I2C, is circuit switched, meaning that there can only be one communication happening on the line at a time. SPI was first used in modular robots in polypod in 1993. Since then, only superbot has used it for intermodule communications. it is a reasonably popular choice for intra module communications, with Odin being the first to use it for this purpose in 2008 and subsequently by CoSMO, Trimobot and Fable.

1-Wire

One wire was introduced by Dallas Semiconductor in the 1990s. 1 wire is an asynchronous bus that operates as an open drain with a pullup resistor. Data rates up to 16 Kbits/s can be achieved with reasonable reliablility. There is no defacto standard defined for the protocols above the physical layer in 1-wire serial communication, It is usually left up to the developer to define a software protocol to use. 1-wire is appealing for applications where speed is not an important factor but IO count on a microcontroller is limited, for example. A common ground must be shared between nodes. 1-Wire has been used for inter module communications in soldercubes in 2016. It has been used for intra module communications in Kairo 3 in 2014 and soldercubes.

UART

UART (Universal Asynchronous Receiver Transmitter) is, as the name suggests an asynchronous communication standard. UART uses a minimum of two electrical lines for communication, Rx and Tx (Receive and transmit respectively) There is also the option to include ready to send, clear to send, data terminal ready, data set ready and device carrier Detect lines. These extra lines can be used for flow control at the hardware layer, although most modern systems implement these functions in software protocols. Commonly used protocols with UART are RS232 and RS485. UARTs typically operate at 9600 baud. A common ground must be shared between nodes. UART has been widely used in modular robot systems due to it's low conductor count and ease of implementation. It was first used by Molecules in 1998 and subsequently by I-Cubes, M-TRAN, Stoachastic, Molecubes, Odin, THOR, Distributed Flight Array, ReBis and Hinged Tetro. It has been used for intra module communications initially by M-Tran in 2002 and subsequently by YaMoR, Molecubes, U-BOT and T.E.M.P.

Ethernet

Ethernet is an asynchronous wired communication standard developed to network computers together. There are many different standards under the Ethernet name, ranging from the first 10MBit/s 10BASE-T standard developed in 1980 to modern 10 Gigabit standards. A common standard is 100Base-TX, which operates at 100 MBits/s. There are two differential pairs required, four individual conductors. It operates at a bandwidth of 31.25 MHz. Implementation of an Ethernet communication system in an embedded device requires a MAC (Media access control) device and an Ethernet PHY (Physical layer) device. Many modern microcontrollers include an internal 100Base-TX MAC peripheral. This device performs packet encapsulation and outputs data in MII (Media independent interface) format which is passed to the PHY IC. The PHY IC performs encoding and line rate conversion for the data to be sent along the Ethernet medium. Ethernet has only been used in one modular robotic system to date, CoSMO, part of the symbricator project.

Summary

A detailed review in to communication protocols used in modular robotic platforms has been carried out. A quantitative review was completed which identified Infrared as the widest used communication strategy in modular robot platforms, taking up a 24.6% share. The second highest was UART, taking up a 15.4% share. There were 17 different protocols identified, showing a lack of agreement or standardisation in the research field as to which protocol is the most appropriate to use.

2.9 Commmunication protocols in IT Systems

In order to create a modular robot implementing a novel communication system, inspiration needs to be taken from other fields of research. This reduces the risk of other protocols that haven't yet been used in modular robotic platforms being overlooked. The most successful field implementing communication is IT systems. This section will review some of the communication protocols that have become ubiquitous in IT systems with focus on higher level protocols that might provide necessary insights in order to ensure the increasingly complex networks of modular robots can communicate effectively.

UDP

Table 2.6: UDP packet header

Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Source Port																Destination Port															
4	32	Length																Checksum															

UDP stands for User Datagram Protocol, which is a transport layer protocol used in computer networking. It provides a simple and lightweight method for sending datagrams, or packets of data, from one host to another.

UDP is connectionless, which means that it does not establish a dedicated end-to-end connection before sending data. Instead, it sends datagrams directly to the destination host, without checking if the host is available or not. This makes UDP faster and more efficient than connection-oriented protocols like TCP, but also less reliable.

UDP is often used for time-sensitive applications where speed is more important than reliability, such as online gaming, real-time video and audio streaming, and Voice over IP (VoIP). It is also used for simple network protocols that do not require the reliability and error checking provided by TCP, such as DNS (Domain Name System) and DHCP (Dynamic Host Configuration Protocol).

One of the benefits of UDP is that it has a small header size, see Table 2.6, which means that it uses less network overhead than TCP. However, it also means that it does not have some of the features provided by TCP, such as flow control, congestion control, and retransmission of lost packets. Applications that use UDP are responsible for handling these issues themselves.

Overall, UDP is a simple and efficient protocol for transmitting datagrams over a network, but it is not as reliable as TCP and may not be suitable for all types of applications.

The packet structure of UDP, also known as the UDP header, contains four fields:

Source Port: A 16-bit field that specifies the source port number of the sending process or application. Destination Port: A 16-bit field that specifies the destination port number of the receiving process or application. Length: A 16-bit field that specifies the length of the UDP packet, including the header and data, in bytes. Checksum: A 16-bit field that is used for error detection. The checksum is calculated over the entire UDP packet, including the header and data.

The UDP header is a fixed size of 8 bytes, so the maximum length of a UDP packet is 65,535 bytes (2Î6 - 1) minus the 8-byte header.

After the UDP header, the packet contains the data payload, which can be up to the maximum packet length specified in the Length field. The data payload can contain any type of information that the sending application wants to transmit, such as audio or video data, DNS queries, or SNMP messages.

The simplicity of the UDP packet structure and the lack of additional control information make UDP a lightweight and fast protocol for transmitting data over a network. However, the lack of error recovery mechanisms can lead to data loss or corruption, so applications that use UDP are responsible for ensuring the integrity and reliability of their data.

TCP

Table 2.7: TCP packet header

<table>
<tr><td>Offsets</td><td>Octet</td><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td><td colspan="8">3</td></tr>
<tr><td>Octet</td><td>Bit</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><td>0</td><td>0</td><td colspan="16">Source Port</td><td colspan="16">Destination Port</td></tr>
<tr><td>4</td><td>32</td><td colspan="32">Sequence number</td></tr>
<tr><td>8</td><td>64</td><td colspan="32">Acknowledgment number (if ACK set)</td></tr>
<tr><td>12</td><td>96</td><td colspan="4">Data Offset</td><td colspan="3">Reserved
0 0 0</td><td>N
S</td><td>C
W
R</td><td>E
C
E</td><td>U
R
G</td><td>A
C
K</td><td>P
S
H</td><td>R
S
T</td><td>S
Y
N</td><td>F
I
N</td><td colspan="16">Window Size</td></tr>
<tr><td>16</td><td>128</td><td colspan="16">Checksum</td><td colspan="16">Urgent pointer (if URG set)</td></tr>
<tr><td>20</td><td>160</td><td colspan="32" rowspan="3">Options (if data offset >5. Padded with 0s)</td></tr>
<tr><td>...</td><td>...</td></tr>
<tr><td>56</td><td>448</td></tr>
</table>

TCP (Transmission Control Protocol) is a transport layer protocol used in computer networking to establish and manage connections between network hosts and reliably transmit data over the network. TCP provides a reliable and ordered data delivery service, and it is one of the most widely used protocols in the Internet Protocol (IP) suite.

TCP uses a connection-oriented communication model, which means that before transmitting data, it establishes a connection between the sender and receiver, and ensures that both parties are ready to communicate. This connection is established using a three-way handshake, which consists of three messages exchanged between the sender and receiver to synchronize and establish the connection.

Once the connection is established, TCP uses a sliding window algorithm to manage the flow of data between the sender and receiver. The sliding window allows the sender to transmit multiple packets of data without waiting for an acknowledgement from the receiver for each packet. The receiver acknowledges the packets it receives by sending back an acknowledgement message, which indicates the next expected sequence number.

TCP provides reliable data transmission by implementing several error detection and recovery mechanisms. These include checksums for error detection, sequence numbers to ensure that data is received in the correct order, and retransmission of lost packets. If a packet is lost or corrupted during transmission, the sender will retransmit the packet until it receives an acknowledgement from the receiver.

TCP also provides congestion control to prevent network congestion and ensure that the network is used efficiently. Congestion control works by dynamically adjusting the sending rate based on the network conditions, such as packet loss and delay.

The TCP header, Table 2.7, contains several fields that are used to manage the connection and transmit data reliably. These fields include:

Source and Destination Port Numbers: These 16-bit fields identify the sender and receiver applications. Sequence and Acknowledgement Numbers: These 32-bit fields are used to keep track of the order of data and ensure reliability. Window Size: This 16-bit field is used to indicate the size of the receive window and manage the flow of data between the sender and receiver. Flags: These 6 control bits are used to manage the connection, including the three-way handshake, flow control, and error detection.

TCP is used for a wide range of applications, including web browsing, file transfers, email, and remote login. Its reliability and ordering guarantees make it a popular choice for applications that require the transmission of large amounts of data over the network.

Problem with the OSI Stack

Within the OSI network stack there are 7 Layers that interact with each other through abstract layer interfaces. Each of these 7 Layers have its own packet or frame structure, which means that by the time data is being sent across the physical medium, there can be 7 different packets encapsulated within each other. Each of these packets has its own header structure, and this can add up to many bytes of headers being sent on the medium. In a system like the world wide web, this system of abstract encapulation is desireable to allow many different standards to be built upon lower level standards. But in robotics this sort of system is not desirable because it adds significant overhead onto the medium, reducing the amount of actionable data being sent. The types of data flow in a modular robotics system can be well enough specified and constrained to enable all data to be encapsulated in one or two levels of packet encapsulation.

2.10 Further Discussion

What are some of the possibilities with modular robotics?

A modular robotic system can be any group of two or more robots that can cooperate together in some connected system to achieve a goal. Mechanically this can be through mechanical reconfiguration, resource-pooling locomotion or manipulation of the system's environment. Modular robots can also cooperate computationally to achieve goals. Tasks that are computationally intensive but parallelizable could be distributed throughout modules in the system with significant compute resources to reduce the processing time. In a system where multiple modules have, say localisation sensors, data from all of these sensors in the system could be autonomically fused to produce a more accurate and precise measurement result. The task of creating a decentralised control system for such resource allocation and task distribution duties is a difficult one. This section would lead nicely on to the reasoning aspect of the system.

Why is there a need for decentralisation?

In modular robotics it is widely accepted amongst the research community that there is a need for decentralisation. This need for decentralisation is particularly paramount when attempting to deploy at scale. One of the main goals that modular robotics is attempting to achieve is resilience against localised failures. If one or ore modules were to become damaged or dysfunctional, the system as a whole should either be able to reconfigure itself to a repaired state by swapping the dysfunctional modules for functional ones. This means that no one module can be critical to the overall function of the system. For if that module were damaged, then the whole system would stop functioning. Although in heterogeneous modular systems perfect decentralisation is not possible, any system level functions should be completely decentralised. These functions include, but are not limited to: localisation, communication, reconfiguration and data storage.

What are scalability issues?

One of the advantages of modular systems is that they are reconfigurable. A system of modular robots could be theoretically configured with 10 or 100 modules. One of the reasons why large scale modular robotic systems have not been realised to date, is that the communication systems that are used cannot cope with the data throughput required for such systems.

2.11 Chapter Summary

A wide range of modular robotic platforms have been reviewed. The connection mechanisms used and the communciation protocols used have been explored. Communication between modules was identified as a major barrier to better systems of modular robots. For modules to cooperate effectively, there is a need for both high throughput and low latency communication. The majority of modular robot systems to date have used Infrared, UART, CAN bus or Infrared diode pairs for their primary communication. While these work well for propagating control commands around a system, they do not work well for dynamically reconfigurable systems with both low latency as well as high throughput requirements. These systems must be resilient to faults and failures as well as providing high communication bandwidth and power throughput to keep systems operating in harsh and challenging environments. Potential applications of dynamically reconfigurable heterogeneous modular robots include, but are not limited to: modular satellites, search and rescue, adaptive locomotion and space exploration.

Chapter 3

Hardware Design

This chapter describes the requirements and initial design decisions for a heterogeneous modular robot platform that can be assembled into a variety of structures through a standardised docking mechanism that provides mechanical and electrical connection between modules. The system must be able to form three dimensional lattice structures. The focus of this research is on physical and electrical communication between modules, hence the main parts of the design include the docking connector and communication subsystem. These two components are closely inter-dependant as the mechanical aspects of the docking connector must not unduly hinder the electrical connection between modules. The MoCube modular robot platform has been developed to meet the following high level goals:

1. Single sided disconnect
2. Genderless docking
3. Capability to transmit power and data between modules
4. Enough on board processing power for advanced computation
5. Conform to a 10cm cubic structure

3.1 Designing an embedded hardware system

With the high level goals of the modular robot system set out, it is now a process of distilling them down in to actionable, functional requirements. The process of setting out requirements for any embedded electronics project is not always easy or straightforward. In this system there are three distinct sets of requirements to set out: mechanical, electrical and software. Each of these sets are very closely inter-dependant; the mechanical design influences the size and shape of the printed circuit boards that can fit inside the robot, and the electronic design greatly influences the software that will be able to run in the system. This also works the other way around; there might be a specific software routine that must be included, and this only runs efficiently on a specific microprocessor/FPGA. This then influences the PCB design, which usually has an impact on the mechanical design. All of this leads to the realisation that embedded system design is never straightforward, requirements have to be carefully considered, and that compromises in particular requirements have to be made. Another important aspect of designing an embedded hardware system is how the project is managed. A good model to use for implementing successful projects is the Project Management Triangle. Also known as the Triple Constraint or Iron Triangle, it is a fundamental concept in project management that represents the interconnected relationship between three key factors that influence the success of a project. These factors are:

- Scope: This refers to the specific tasks, deliverables, features, and objectives that define what needs to be accomplished in the project. It outlines the work that must be done to achieve the project's goals. Changes in scope can impact other aspects of the project, such as time and cost.

- Time (Schedule): This factor represents the duration or timeline within which the project must be completed. It includes setting deadlines for various milestones, tasks, and the overall project completion. Changes in time can affect scope and cost as well.

- Cost (Budget): This refers to the financial resources allocated to the project. It includes all expenses related to labor, materials, equipment, and other resources required to complete the project. Managing costs effectively is crucial to ensure project feasibility.

These three factors are interconnected, and changes in one factor will often impact the other two. The central idea of the Project Management Triangle is that these factors are interdependent, and making changes to one factor will invariably affect the other two. For example: If the scope of a project is increased (more features or tasks are added), the project may take longer to complete, and the cost might go up due to increased resource requirements. In a project where there time requirements are hard set, scope will have to be managed relative to the allowed time. If the budget was to be increased, some of the manufacturing work could be outsourced to increase the scope for the given time frame. These factors need to be balanced to ensure that the project is delivered successfully. Any changes to one factor need to be carefully assessed for their impact on the other two factors. Effective project management involves making informed decisions about trade-offs among scope, time, and cost based on the project's objectives and constraints.

3.2 Design Requirements

As stated in section 3.1, setting out the requirements for an embedded hardware system is not always a straightforward process. The requirements set out the functional aspects of the system, without necessarily specifying how they will be implemented. With a complete list of requirements for a project, any competent person should be able to recreate a compatible system following their own methods. Generally, requirements are first set out at a very high level and then honed down closer to the implementation stage.

The key design requirements for MoCube are:

1. **It should be constructed using 3D printing** - Any mechanical aspects of the system should be able to be constructed with the 3D FDM printers available at the University of York. Any manufacturing methods that include machining metal parts or injection moulding etc would greatly increase the budget required as this would have to be outsourced.

2. **Modules should be able to perform single sided connection and disconnection** - In order to achieve a fault tolerant system, modules must be able to perform single sided disconnection operations, i.e. one module should be able to disconnect itself from a neighboring module that has stopped operating correctly. This will also allow the system to utilise passive connectors, reducing complexity of some modules where appropriate.

3. **Module docking connectors should be genderless** - The connection mechanism between modules should be genderless, or hermaphroditic. This means that any module face should be able to connect to any other module face of the same design.

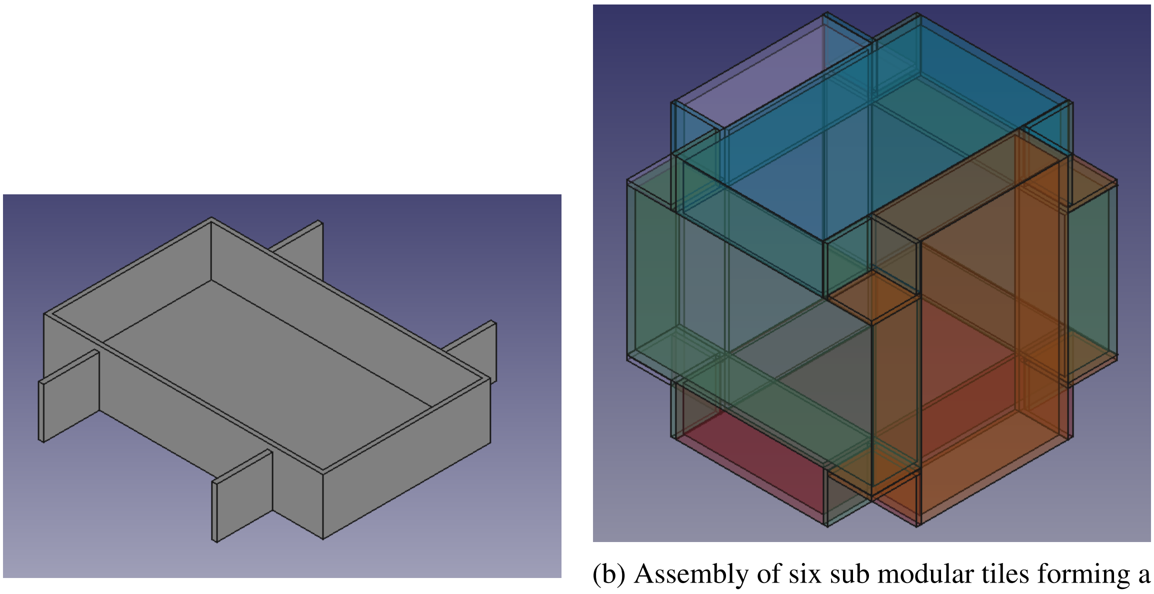
4. **Connectors should not require steady state power to hold connection** - The docking mechanism should not reqiure a constant power source to hold a connection, i.e. the module-to-module connector should only require power for transitioning to an active connect to active disconnect state.

3.3 Design Overview and Features

The Mo* modular robot platform has been designed to demonstrate a high bandwidth communication strategy between modules in a heterogeneous modular robot system. Through the literature review, it was found that only one other modular robot,CoSMO[22] to date has used a communication system based on IEEE 802.3 Ethernet, and as such, implemented communications beyond other standards designed for low data rate, low node count networks.

Module Form Factor

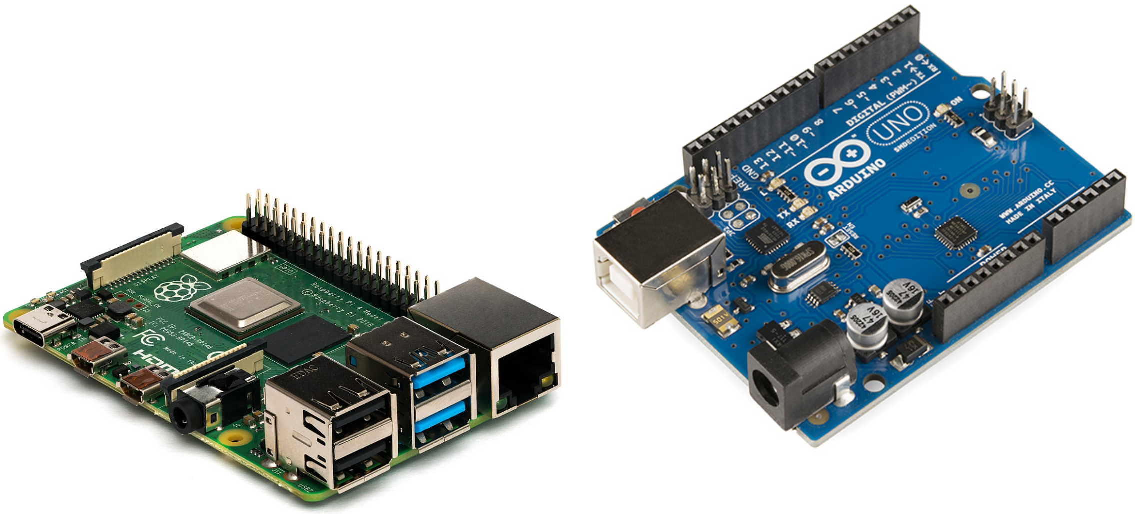
The first mechanical design decision to be made is the form factor, or shape of, the modules in the system. There have been many different module form factors in previous systems, outlined in the previous works section. One of the goals of this project is to create an extensible platform that can be built upon in the future. As such it is useful to remember to keep any design decisions as simple and flexible as possible, as the form factor of the modules will greatly influence how adaptable the system is. It should firstly allow modules to connect to each other to create a solid structure, so it should be a space filling polyhedron. The outer shape of the module should also form a platonic solid, one which the faces of the polyhedron are congruent, and each vertex has the same number of faces connected to it. Conforming to these requirements will allow any module face to connect to any other module face in any orientation. The only space filling platonic solid that exists is a cube. Hence, this is the shape used as the basis for Mo* module design. Expanding on the desirable trait of extensibility, and to increase the likelihood of researchers taking the concepts developed in this work further and developing their own heterogeneous Mo* designs, it is useful to create a high level standard for how modules are designed and constructed, while allowing customisation. Conforming to the theme of modularity, a standard sub modular tile architecture was developed. This sub-modular tile design allows a cubic module to be constructed from dimensionally-identical tiles. A representation of this tile design and how is constructs in to a cube is shown in figure 3.1 Each tile in this case forms one side of the cube shape. If the dimensional properties of a tile conform to a unified standard, then any tile can be constructed into any module, thus reducing the need to reproduce full modules for small feature extensions, such as upgrading sensors or actuators. This also opens up the opportunity to make upgrading modules much easier. One could envisage a scenario where a catalogue of module tiles exist that could be bought, or built from open source designs, allowing researchers to accelerate the path to creating heterogeneous systems tailored to their own specific area of research. This could potentially greatly increase the rate at which modular robot research is advanced, with researchers not being required to create a whole system from scratch only to test out a novel reconfiguration algorithm, for example. With the shape defined, the size of a module is the next property to determine. As the goal of the project is to create a extensible system, it would be useful if future researchers could use off the shelf microcontroller and microcomputer boards in their modular systems. This would greatly accelerate the development of new tiles to create new heterogeneous modules. Two of the most popular development boards worldwide are the Raspberry Pi [84] and the Arduino Uno [85].



(a) Single sub modular tile design

(b) Assembly of six sub modular tiles forming a cube

Figure 3.1: CAD concept designs of a cubic module and its sub modular tiles.



(a) Raspberry Pi 4 SBC

(b) Arduino Uno Development Board

Figure 3.2: Popular development boards that could be integrated into modules.

If the modules can accommodate these development boards, most computational requirements for new tiles would be covered. The latest model of raspberry pi available at the time of writing is the Raspberry Pi 4 Model B [86], seen in figure 3.2a. This single board computer (SBC) contains a Broadcom BCM2711 64-bit quad core processor, 1GB-8GB of RAM, and a variety of I/O peripherals. Rasberry Pi's have become almost ubiquitous in embedded hardware prototyping in academia, especially in robotics. The dimensions of a Raspberry Pi SBC are 85mmx56mm. Arduino produce a range of boards in a couple of different form factors. The most popular of these is the "classic family"[87], seen in figure 3.2b. This family includes the Arduino Uno model, their best selling to date. The dimensions of all boards in the classic family are 68.6mmx53.3mm. If these boards are to be accommodated in this modular platform, inner dimensions of the module should be at least 90mmx60mm. As a cubic structure has been identified as the required shape of the module, the dimensions of a tile will need to be equal. A 100mmx100mm outer tile dimension would give a 100mm^3 module size.

Docking connector

The docking connector is the subsystem of a modular robot that gives each module the ability to connect and disconnect with other modules in the system, allowing sets of modular robots to form structures and kinematic chains. One constraint has already been set out in this section. That is that the each tile of the cube shall be 100mmx100mm outer dimensions. The docking mechanism must be able to fit inside of these dimensions. The depth of the docking connector is an important factor to consider, as if the connector takes up too much space on the inside of the module, there will be less room for electronics, batteries etc. As stated at the beginning of this section, the connector should be hermaphroditic, have four rotational degrees of symmetry, allow power and data transmission between modules, and should have the ability to perform a single sided disconnect. For the connector to be hermaphroditic, it needs to have an equal and symmetrical number of male and female components. These could be pins and latch holes, keys and keyholes, opposing polarities of a magnet or hooks and latch holes. For the connector to be four times rotationally symmetric these alternating components need to be arranged in a rotationally symmetric manner. This design style of alternating male-female components has been used widely for module docking connectors in modular robot platforms such as HyMod [12], CONRO [30], M-TRAN [40] and others. Where docking connectors have protruding components that cannot be retracted, the reconfiguration capability is reduced. A module cannot be translated linearly into place between two opposing modules if there are protruding components. For this reason it is desirable to have no protruding components from the docking connector, or if there are any, they should be retractable. It would be ideal to have no components in the docking connector that require retraction, as these will need to be retracted to the interior of the module, leaving less space for module electronics and other components. The first design that was explored involved a circular array of nut and bolt like structures. There are 4 bolts and four nuts in a circular pattern, giving the four rotational degrees of symmetry that are required. The nuts were to be actuated by a motor meshed to each of the drive heads of the bolts. A concept CAD design of this mechanism can be seen in Figure 3.3

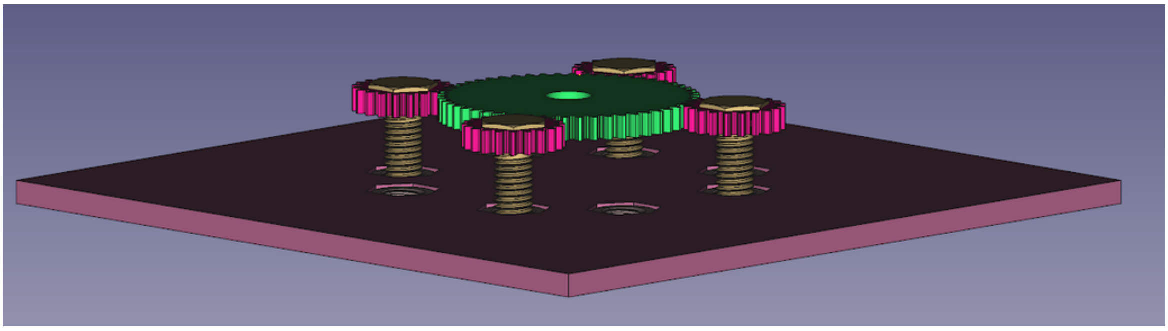
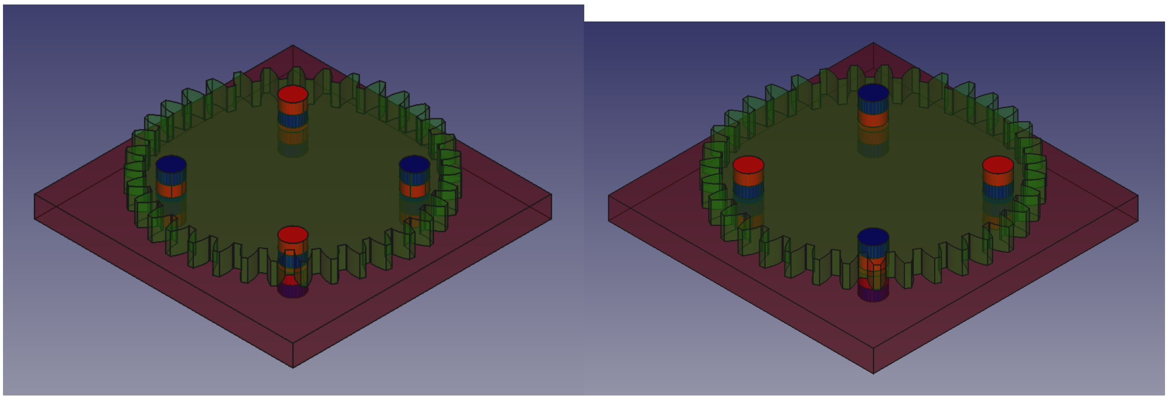


Figure 3.3: CAD Sketch of a single sided bolt connector

This sketch shows the idea of a bolted connector. Advantages of a design like this include a very high strength bond between modules, and actuation is only required during a transition phase. As long as the nuts and bolts were made of a conducting metal, they could also be used for electrical connection between modules, giving eight separate channels. In order to make this design work as a single sided disconnect mechanism, the nuts would need to be actuated as well as the bolts. If the neighbouring module had locked up, that would be the only way to disconnect the two modules. Another disadvantage of this design is that there is very little misalignment tolerance, the bolts would have to line up precisely with the nuts on the opposing module. As well as this force needs to be translated from a motor to the drive gear, pictured in green. As this gear needs to translate linearly with the bolt gears, there would have to be provision for linear translation of the coupling between the motor and the gear, or the whole motor assembly would have follow the gear as one unit. For these reasons, it was decided that the mechanical complexities of this design would become too great to implement this connector design within the scope of this project. If someone could get this design to work, it could be a very competent addition to the field of modular robots.

While the bolt docking connector design did not end up with something implementable, a lot of lessons were learned about some of the nuances of docking connectors. Trying to keep the underlying principles of the bolted design, magnetic coupling mechanisms were explored. Using magnets would eliminate the need for any protrusions in the connector as they could be embedded flush with the surface of the docking connector. Could there be an analog for this style of connector, but using magnets for the coupling force? The first option is to use electromagnets. This, however does not conform to the requirement that power should only be required in a transition state. The second option considered was semi-permanent magnets. These are a good looking option for a docking connector. Semi permanent, or switchable magnets only require power when transitioning from a magnetised state to an demagnetised state, or vice versa. Two modular robot platforms that have implemented switching permanent magnets are Miche [54] and Pebbles [61] both developed by Kyle Gilpin at MIT. In Miche, an off the shelf rotary magswitch [88] was used and actuated by a DC motor. In Pebbles, a electroswitching magnet was constructed from a Neodymium-Iron-Boron magnet coupled to an Alnico V Magnet wrapped in copper wire. Each of these magnets have the same magnetisation strength, but the Alnico magnet is a lot less coercive, meaning that around 100 times less applied magnetic field is required to switch its polarity. This means that the device as a whole can be switched to a state where the magnet poles are aligned, roughly summing their magnetic attraction forces, or switched to where the magnetic poles of the two magnets are anti-polar to each other, resulting in a near net-zero magnetic force. Each of these methods are attractive options, although there are very few commercially available semi permanent magnets at the size scale that would be required for the system described in this work. Manufacturing either of these solutions would be possible, although it would greatly increase the time required to design, source parts and assemble a docking connector. Being able to create this sort of connection mechanism out of readily available, off-the-shelf permanent magnets, without requiring too many additional manufacturing steps would be better. Taking inspiration from the rotating semi-permanent magnet design, a concept was explored where there are two discs of magnets, one behind the other. The concept design is illustrated in Figure 3.4.



(a) Active Connect State (b) Net-Zero Force State

Figure 3.4: Concept design of a docking mechanism based on a rotating array of permanent magnets

Figure 3.4a shows the connection mechanism in the active connect state, where the polarity of the magnets are aligned with each other. Figure 3.4b shows the connector in the net-zero, or passive state, where the magnets are arranged anti-polar to one another.

The connector design was promising. Unlike the first concept, there is only one degree of freedom required. The magnets would also mean that the connection mechanism would cope well with any misalignment problems. A problem with this design is that there would be significant mechanical stresses presented on to the driver gear when the connector is in the passive state. Also, the concept design shown with four magnet positions on the face would only be two times rotationally symmetric. To achieve 4 times rotational symmetry, 8 magnet positions would need to be used. That would total 16 magnets per face. On a cubic module with 6 active connection faces, that would yield 96 magnets in a module. While this would be achievable, it would be much more desirable, from a mass, cost and implementation perspective, to have a maximum of eight magnets per face.

The core premise of the concept design shown in figure 3.4 is that a switchable permanent magnet can be created by rotating two permanent magnets relative to each other. If this relative rotation between magnets could be moved from inside a connector to between connectors, then the same effect could be achieved with the desired 8 magnets rather that 16 in the previous design. The problem with doing this is that a net zero rotation force between two modules needs to be achieved. This means that rotating the connector mechanism will not exert any net force on the connected module during actuation. If the whole ring of magnets was rotated as one, radially about the center of the face, then one module would rotate relative to its neighbour. One way to achieve net zero rotation force would be to have two concentric rings of magnets, each rotating in opposing directions. For this to result in an absolute net zero rotation, the magnets on the inner ring would need to be stronger than the magnets on the outer ring, so that each ring exerts the same torque about the center of the connection face. To obtain magnets of the right relative strengths would be challenging. The method of solving this relative rotational force problem can be solved if a concentric ring of 8 magnets is subdivided into four pairs of magnets with opposing polarities at a normal to the tile surface, each pair rotating about their own independent axis. This is visualised in Figure 3.5.

If two neighbouring modules are in the active connect state, then each magnet pole on the face will be co-axial with the opposing pole on the other face, creating an attraction force and holding two modules together as a function of the sum of the attraction force between the magnets. If one module transitions to the active disconnect state, then each magnet pole on the face will be co-axial with the identical pole on the other face, creating a repelling force and disconnecting two modules.

This system can be modeled as four diametrically polarised magnets. If each of these diametrically opposed magnets are rotated about their own centers, there is no rotational axis

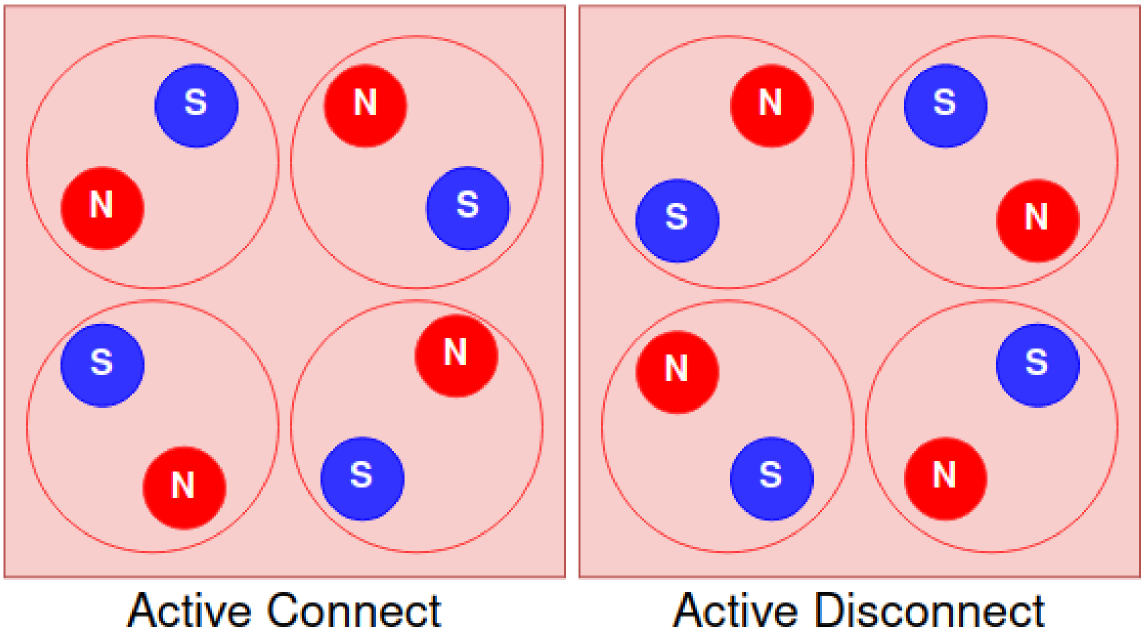


Figure 3.5: Concept of the rotating pairs of permanent magnets

in the center of the docking connector hence there will be no net rotation force exerted on one module by another.

A third ’connection state’ can also be achieved with this connector. In this state the magnet pairs are arranged such that a line drawn between the two magnets in each pair is at a normal to the corresponding magnets in the neighboring module’s connector. This is shown in Figure 3.6 In this state none of the magnets in the connector are co-axial with each other, In effect this creates a net-zero attraction force between two connecting modules. This can be viewed as a passive state. There is also no combination of translations and rotations of one module relative to another that will align sufficient magnet anti-poles to create a net attraction force between modules.

This connector concept requires rotational force to transition between connection states. This could be provided in a number of ways. The initial concept was to have a small DC servo, such as the TowerPro SG90 [89]. This could directly couple to the back of one of the magnet pairs, and radial gear teeth could be used to mesh to the other 3 pairs. Two of the gear pairs would rotate in the opposite direction to the others. This method of actuating the connector mechanism had to be abandoned as a servo small enough could not be identified that would have the required torque output to actuate a connection mechanism with sufficiently strong magnets. A larger servo could be used, but this would impact the space available inside a module too greatly.

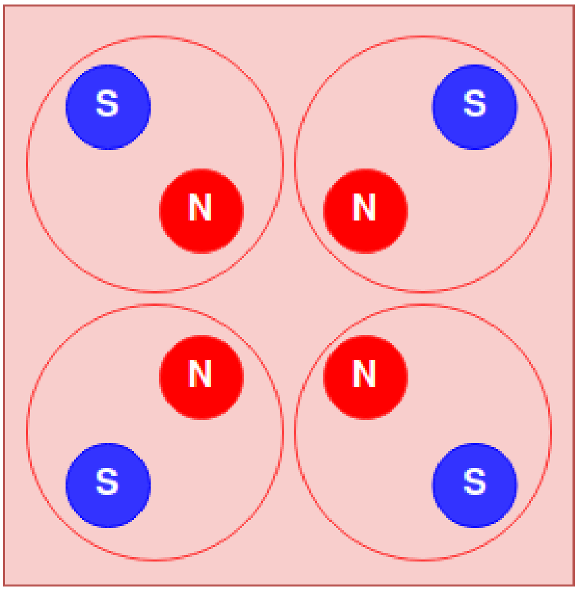


Figure 3.6: The arrangement of magnets in a passive connection state.

The second iteration of this design utilised a central driver gear, which meshes to each of the magnet pairs. The central driver gear could then be coupled to a micro metal gear motor, such as [90]. These motors come in a range of power ratings and have gear ratios from 5:1 to 1000:1. A maximum stall torque of 12 Kg/cm can be achieved with these devices. This gives flexibility when tuning magnet strength to motor output torque. Speed of connection for state transition is not a hard requirement of this system, so that can be sacrificed for sufficient torque to actuate the mechanism.

A mathematical model of the connection mechanism can be created to estimate the motor output torque required for a given magnet strength and inter-magnet spacing. A graphical representation of this is given in figure 3.7. Magnets sourced from suppliers such as [91] are specified with a 'slide resistance'. This is given in Kg, i.e. the mass required to hand from one magnet to slide it off another magnet, at a normal to the axis of magnetism. This is denoted as F_m. The distance of the magnets in a pair to the axis of rotation is an important factor to consider. All eight magnets on the connector must be spaced uniformly in a circular array around the connector or the rotational degrees of symmetry will not be maintained. If the inter magnet spacing is reduced, the magnets will all be closer to the center of the connector. This will reduce the holding force of the connector against bending moments between two modules. Ideally the magnets should be spaced as far apart as possible. The torque τ_p required to overcome the slide resistance of the two magnets in that pair and initiate

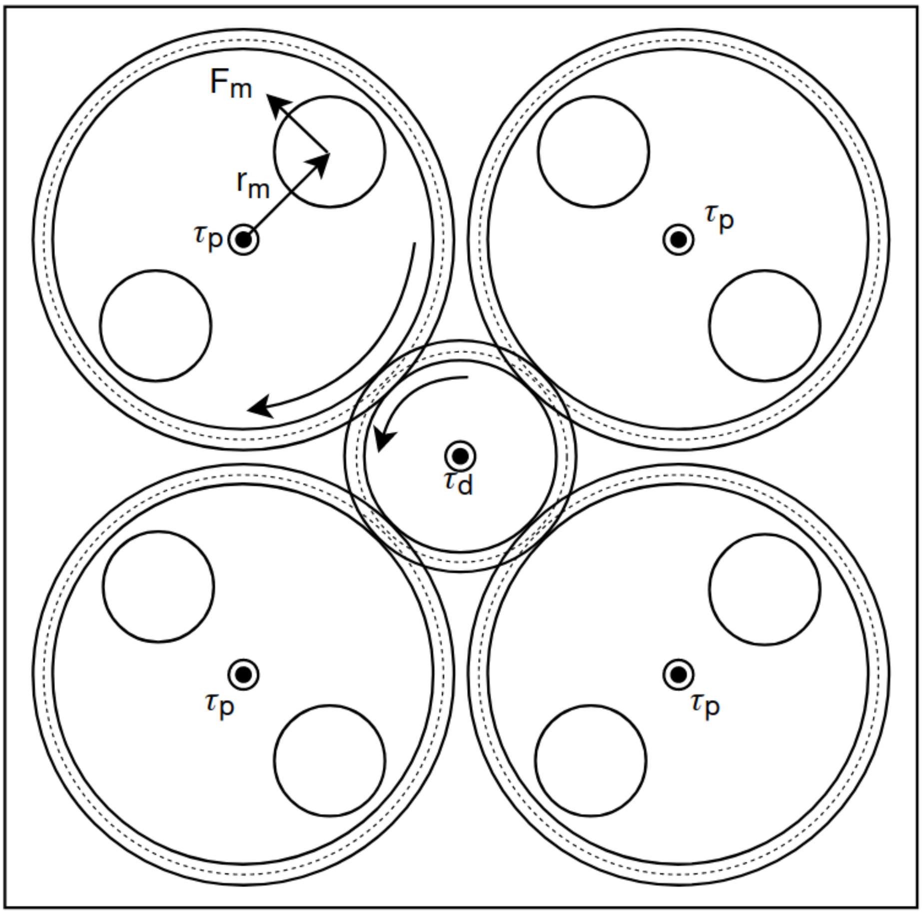


Figure 3.7: Schematic view of the docking connector concept.

a transition between connection mechanisms is given as

$$\tau_p = 2F_m r_m \tag{3.1}$$

where τ_p is the torque in $Kg.cm$, F_m is the slide mass-resistance in Kg of the magnet, and R_m is the turning radius of the magnets, given in cm. The motor drive torque required to actuate the connector is given in equation 3.2

$$\tau_d = 4i\tau_p \tag{3.2}$$

where i is the gear ratio between the driver gear and the magnet pair. Eq. 3.1 can be substituted in to Eq. 3.2 to give

$$\tau_d = 8iF_m R_m \tag{3.3}$$

which gives the driving torque required as a function of the slide resistance of the magnets and the spacing between them. This a simplified model of the system, as it does not account for any of the mechanical friction in the rotating components. This should not cause the model to be an inaccurate estimate as the force exerted on the system by the magnets is substantially greater than any force from mechanical inefficiencies, given a good design implementation.

The mechanical holding properties of the connector is an important factor to consider. The connector must be strong enough to hold other modules. There are four modes of forces that can be exerted between two modules:

- Axial
- Twisting
- Bending
- Shearing

These are depicted graphically in figure 3.8. The axial force is the sum of the axial holding

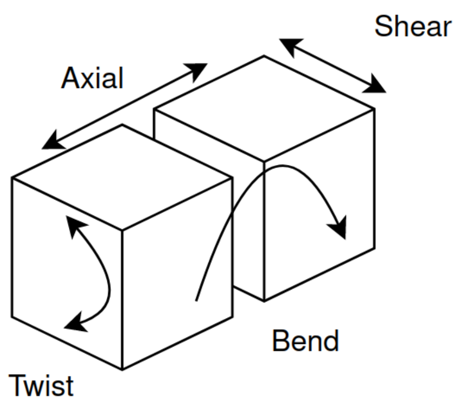


Figure 3.8: The modes of force that can be applied between two modules.

force of all eight magnets. The shearing force, which can act at any angle, is the sum of the shear resistance of all of the magnets. The twisting force is the same as the shearing force but acted on as a torque. The bending force is a bending moment exerted around a pivot point anywhere outside the 3D space filled by the modules. The resistance to the bending force is close to the resistance of the axial forces. The shear force between two magnets is smaller than the axial force, as it is a function of the axial force and the coefficient of friction between the two magnets. The coefficient of friction of nickel-on-nickel, the material used to coat neodymium magnets, is 0.53[92]. The shear holding capacity of one module relative to another will be equal to roughly half of the axial holding capacity for this design.

The requirement of providing electrical connection between modules must also be fulfilled. This can be achieved by utilising the nickel plating on the magnets. Nickel has a resistivity of $6.99 \times 10^{-8}\Omega/m$. This is two and a half times greater than aluminium at $2.82 \times 10^{-8}\Omega/m$, and four times greater than copper at $1.68 \times 10^{-8}\Omega/m$. The length of the conductor between modules is equal to two times the height of the magnets. This is of the order of 10mm, so the higher resistivity of nickel will not have a great impact on the channel impedance. Electrical contact can be made with the inward side of the magnets with spring loaded contact pins such as [93].

Electrical Connection between modules

The next part of the system to consider is the communication between modules. It was identified in chapter **??** that 100Mb Ethernet is a good candidate for the physical layer of communication between two modules. There are eight independent electrical channels available in this connector design. If the relative orientation that two modules connected to each other with was a constant, then there would be eight channels available to use. With eight conductors available, IEEE 802.3 Gigabit ethernet could be used, as it consists of four data pairs. One of the specifications of this system, however, is that two modules must be able to connect to each other in any four relative rotations. The connection mapping between two modules will not be identical for any given orientation. Figure 3.9 provides a physical view of how the conductors will be aligned for all rotation permutations. The conductors are labeled the same in each of the modules. Notice the magnets in module B are mirrored as the diagram is looking at the inside of its face. There is an arrow on each of the module B permutations depicting the current orientation. Module A's orientation stays constant. The connection permutations that this exhibits are depicted in a schematic view in figure 3.10. Each instance depicts a relative rotation between two modules. From these schematic views it can be difficult to discern any pattern in the way the connectors relate to one another. There is a pattern that is very much worth noting however. The contacts (A-C-E-G) on one side only ever connect to (B-D-F-H) on the other side. This holds in both directions. If the contacts were labelled numerically around the connector, as in a clock, then the odd numbers on one face would connect to the even numbers on the other. With this information, this schematic of the connection permutations can be simplified into two distinct states by grouping some of the connectors together. This new grouping is shown in figure3.11. One permutation is for 0 and 180 degrees relative rotation between modules, and the other is for 90 and 270 degrees

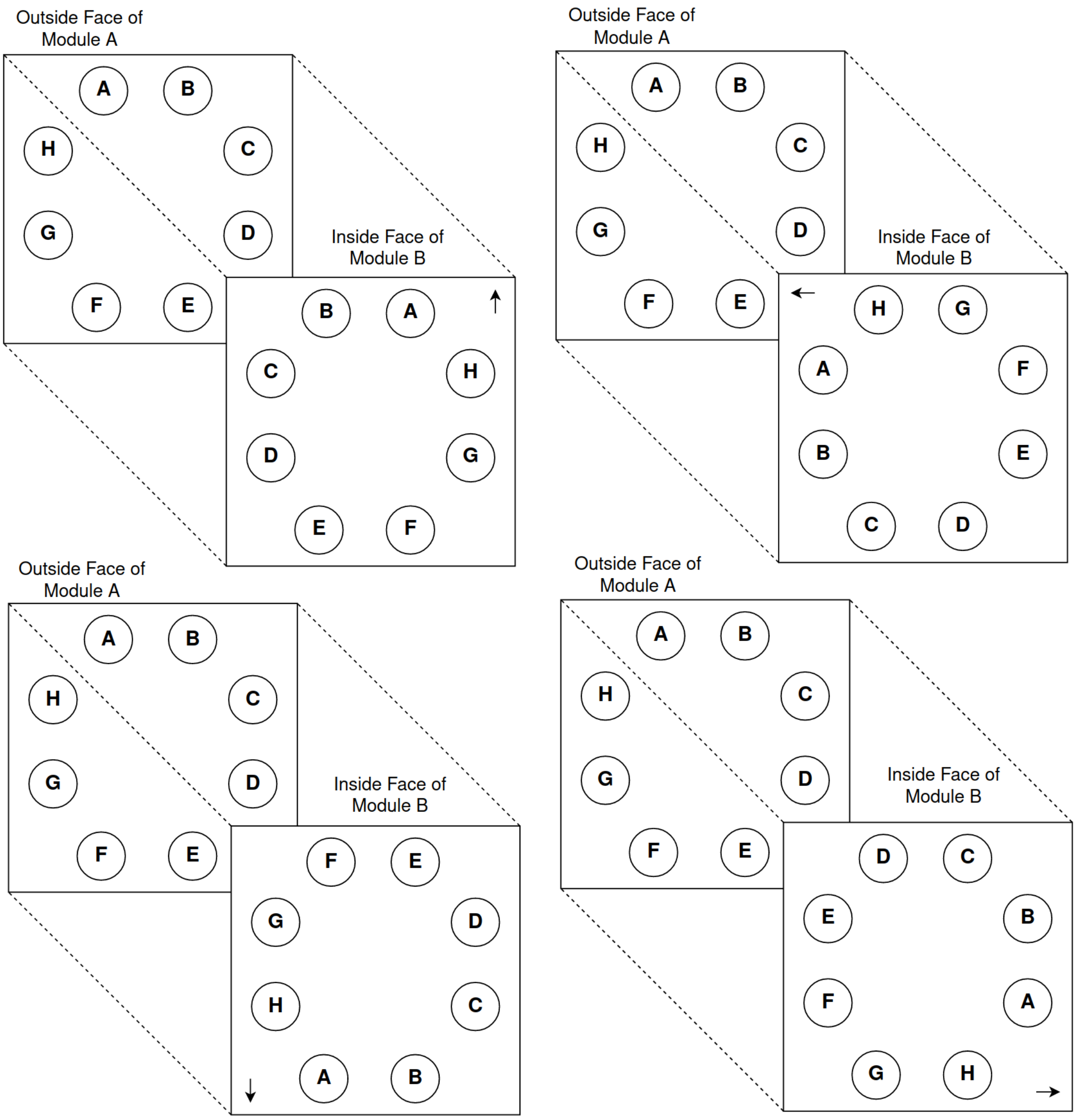


Figure 3.9: Conductor alignment for four rotations.

relative rotation. Each of the pairs in this schematic are diametrically opposite contacts in

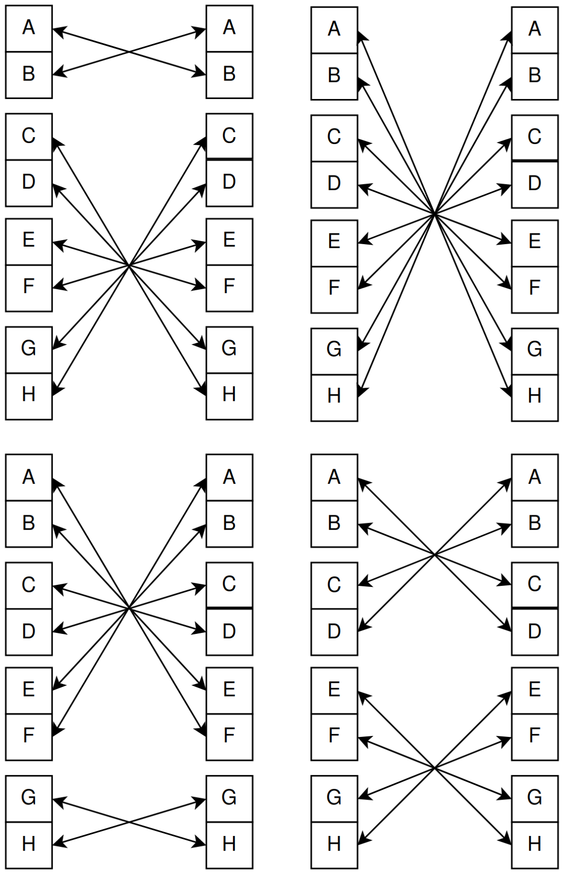


Figure 3.10: Schematic view of connection permutations.

the connector. With this it can be stated that if opposing contacts in an eight contact radial array connector are combined, two electrical connection permutations can be achieved. This connection permutation simplification is only meaningful if it can be leveraged. There is a feature of IEEE 802.3 100Base-TX Ethernet physical layer that was designed in to the standard to allow backward compatibility between devices that used either MDI or MDI-X. This relates to how the transmit and receive data pairs are connected, the transmit data pair must connect to the receive data pair. Typically, MDI is used on end-points and routers, while MDI-X is used on hubs and switches. Historically, there were two conventions for wiring an Ethernet cable to account for this. In 1998 Hewlett Packard submitted a patent "Apparatus for automatically configuring network media connections"[94], which was a technology to detect crossover in the Ethernet PHY IC and account for it automatically. This eliminated the need for crossover cable wiring, allowing either type of wiring to work with any two compatible Ethernet PHYs.

For this Project, this auto-MDIX technology means that two compatible Ethernet PHYs will successfully connect and communicate in any relative orientation between connecting modules, given that the opposing magnets on each face are connected to the same data line. This is graphically shown in figures 3.12 and 3.13. Figure 3.12 shows two modules in 0 and

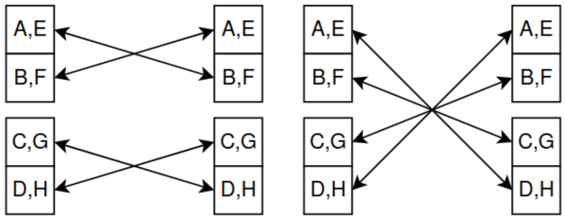


Figure 3.11: Schematic view of the simplified connection permutations when opposing contacts are paired together.

180 degrees relative rotation from each other. In these rotations, and with this conductor arrangement, the Ethernet channel is connected in a straight through state in these rotations. That is, the TX data pair is connected to the RX data pair in each direction. Figure 3.13 shows the connection permutation when two modules are connected in 90 and 270 degrees relative rotation. In this state the transmit pairs are connected to each other, and the receive pairs are connected to each other. In this state the Ethernet PHY will make use of the automatic crossover detection.

Module Electronic Subsystem

With the mechanical properties and communication system set out, an electronic system needs to be designed to support these capabilities. From the work done previously in this section, the following constraints are imposed on the electrical design.

The electronic subsystem must:

- Fit inside a 10cm x 10cm x 10cm cube using the sub modular tile design discussed.
- Provide capability to control the actuation of the docking connector on each face.
- Provide communications capability on each side of the cubic module.

With these requirements in mind, some useful design decisions can be made. The system must provide actuation capabilities to each of the docking connectors on the module, which could total up to six. There is no feasible way to selectively mechanically transfer power between tiles, so each tile will need its own electromechanical actuation. This can be provided by a micrometal gear motor as mentioned previously. The system must be able to actuate

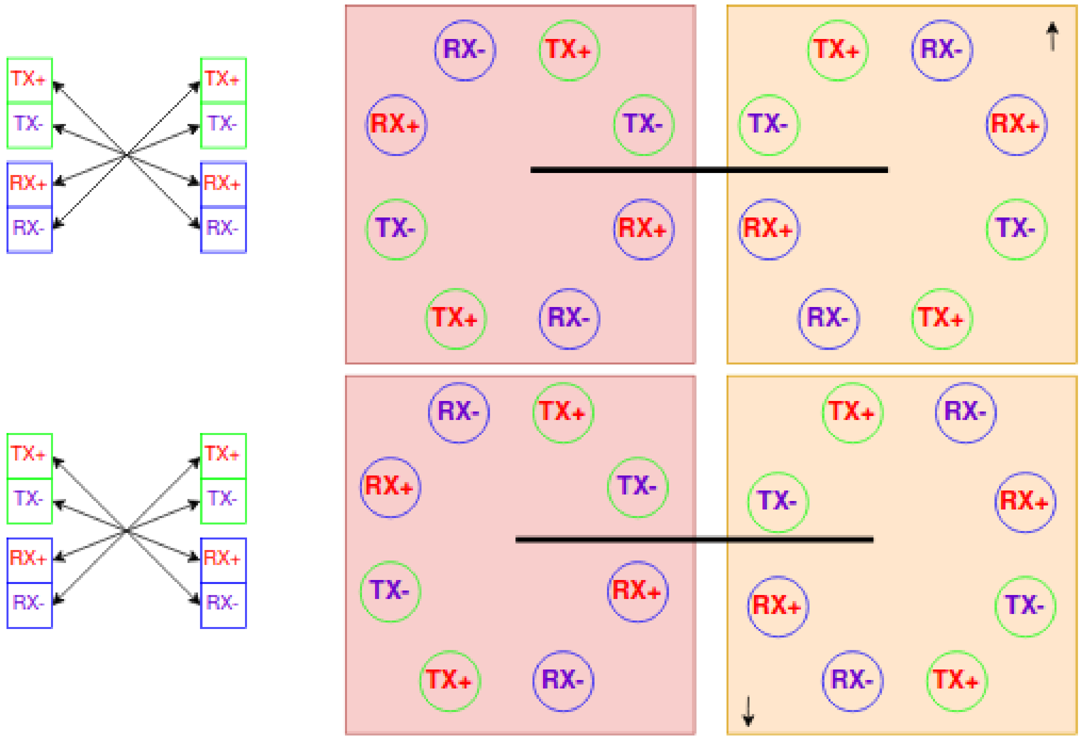


Figure 3.12: Schematic and physical view of connection permutations for 0 and 180 degrees relative rotation.

more than one connection mechanism at the same time, as this scenario could arise during many reconfigurations. This means that six motors and motor drivers will be required in the electromechanical subsystem. Feedback on the current position will also be required, this can be provided by a rotary encoder attached to the backshaft of the micro metal motor. A rotary encoder only provides relative position information. Absolute position information is also required to allow the system to determine where the connector is in its rotation cycle. The implementation of this is discussed in the next chapter.

The next crucial part of the electronic subsystem design is the communications. 100Base-TX Ethernet has now been specified as the hardware layer for the communications between modules. 100Base-TX (hereafter referred to as just Ethernet) requires a line driver for the physical layer integrated circuit to interface between the line side of the Ethernet channel and the controller side. These ICs implement the hardware layer specification features such as Auto MDI-X that are useful in this project. In order to have the capacity to communicate on all six faces of a module at any one time, there will need to be six of these Ethernet PHY ICs on board each module. Any form of multiplexing would not work in this scenario as there

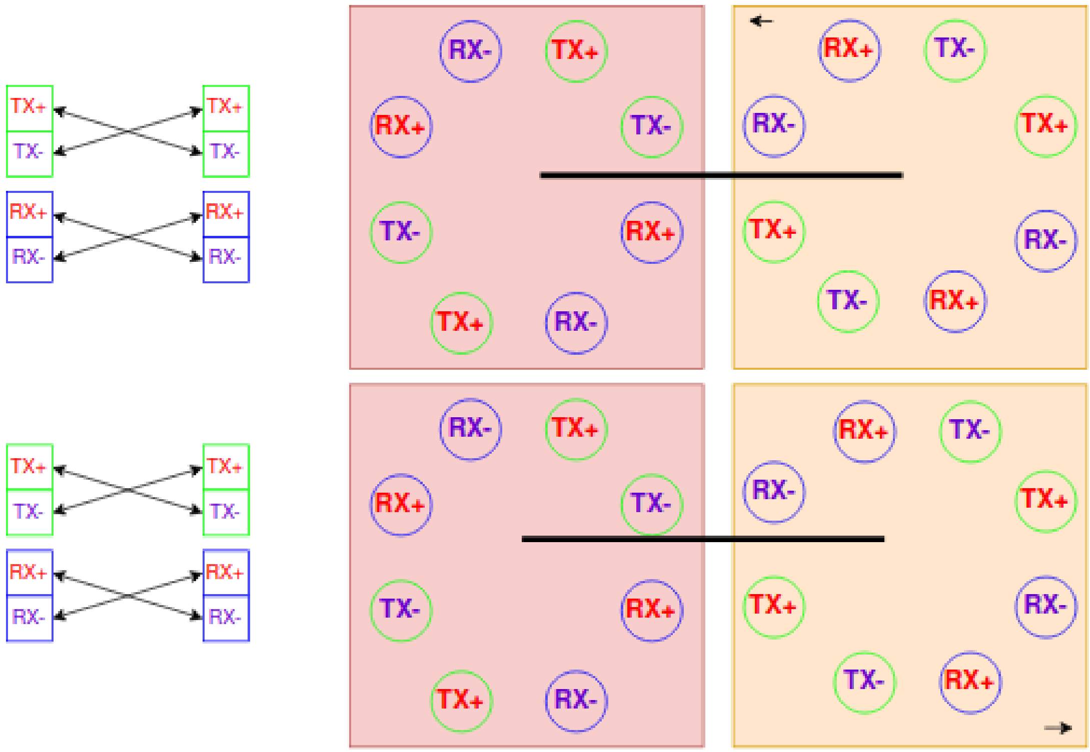


Figure 3.13: Schematic and physical view of connection permutations for 90 and 270 degrees relative rotation.

could be multiple routes processing traffic at any one time. There are a plethora of Ethernet PHY ICs on the market to choose from, the product selection will be discussed in the physical implementation chapter. In order for a software communications protocol to be developed on top of this hardware layer for reconfigurable modular robot systems, a processor is required. This must have the capability to interface with six PHY ICs over the industry standard RMII interface. It must also allow the flexibility to implement custom protocols. While many microcontrollers on the market have ethernet medium access controller peripherals built in to them, and in some cases up to four MACs to one chip, these are constrained to using industry standard protocols and offer little flexibility. For this reason an FPGA is a good candidate for this purpose. FPGAs offer the most flexibility in designing hardware interface systems, and with the inherent parallel capabilities, offer potential benefits in designing efficient protocol implementations. A processor will still be required in a module to provide the application layer processing, but the FPGA can handle all of the inter module communications. This also creates the potential to make a system where system designers could use something like a "MoCube interface board" that would handle all of the module specific communications and connector control, and use industry standard interfaces to allow designers to easily integrate a variety of processors into modular systems. There are 3 ways to implement an FPGA and processor based design: A single FPGA with a soft core processor, a discrete FPGA and processor combination, or a system on chip (SoC) design that combines FPGA fabric and a processor architecture on to one silicone device. The first option is not particularly feasable for this project, as the size of FPGA and speed grade of FPGA required to implement a processing architecture comparable to discrete processors would make hardware implementation complex and increase the scope of the project beyond the constraints imposed on it. A discrete FPGA and processor combination is a desirable solution, as this would aid the customisability of the system. This method was initially the desired path to follow, although it had to be abandoned as there was an international shortage of discrete FPGA devices when development was underway. Nothing could be done to get around this. There was no way to get an FPGA device that would suit the requirements of the project within time and budget. This leaves the last option, of an SoC. SoCs are desirable as they are highly integrated devices, which will save on physical space in the module. The system architecture that would be employed is shown in figure 3.14.

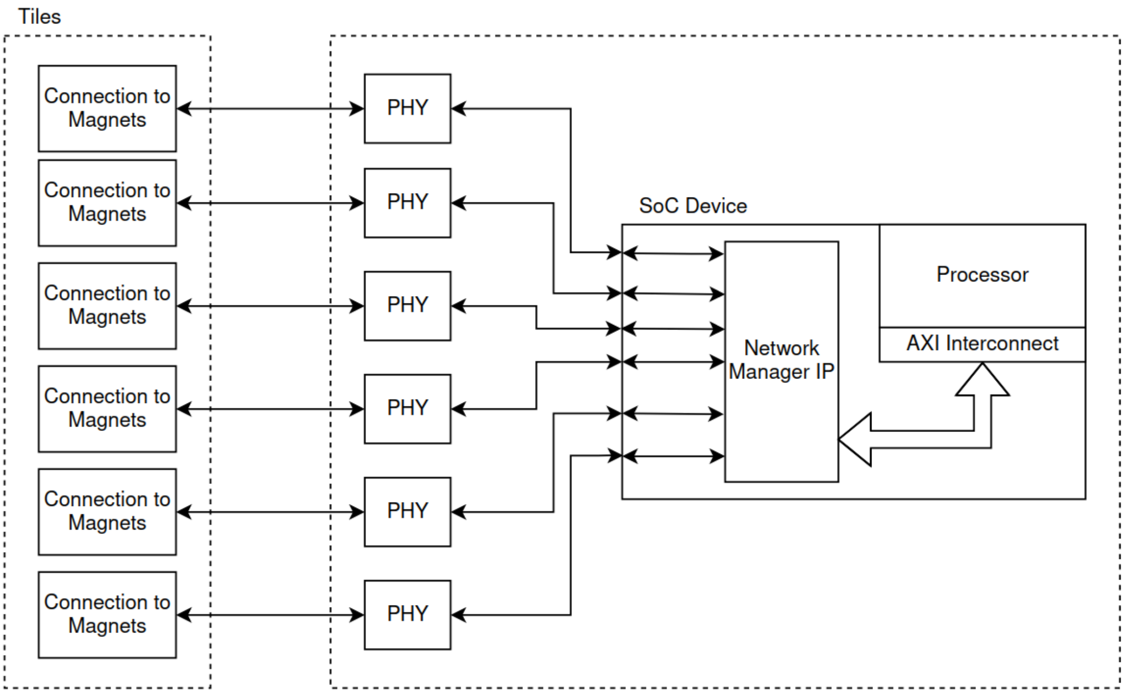


Figure 3.14: High level architecture of a system with 6 ethernet peripherals, and an SoC

Implementation of all of these components on to one board that would fit on to a single tile in the module would be a challenge. It is useful to subdivide the system in to a 'master' board, or tile, then slave tiles. This also enhances the modularity of the system. On the master tile there would be the SoC, Ethernet PHY's and Motor driver for the tile that the master PCB is mounted to. Slave boards would have a small microcontroller for motor control, motor driver, and line side Ethernet discrete components. Power, local communications and ethernet needs to be routed from the master board to each of the slave boards. Since the slave board microcontroller is only responsible for controlling the motor on that tile, a low speed bus based communication protocol can be used for this purpose.

3.4 Chapter Summary

This chapter has summarised the hardware concept design of a heterogeneous modular robot platform. The requirements of the connection mechanism between modules has been set out, then a system has been conceptualised around this to fit those requirements. A module form factor has been discussed that accommodates a sub modular design methodology. Docking connector design has been discussed through the various iterations that were investigated. The final rotating circular array of magnets design has been investigated and analysed both for mechanical and electronic connection between modules. This design allows two modules to connect with four degrees of rotational symmetry, maintaining maximum magnetic attraction force, and electronic connection between modules. Finally, the electronic subsystem required to accommodate this module design has been specified and discussed. Implemented, this system will accommodate all of the requirements of a heterogeneous modular robot system that can reconfigure dynamically.

Chapter 4

Physical Implementation

4.1 Introduction

This chapter discusses the final design and implementation of the Mo* modular robot platform. The connector mechanism implementation is discussed, followed by the electronic system of a master tile and slave tile. The embedded software developed for the system is also discussed.

4.2 Sub Modular Tile

The core of the Mo* modular robot platform is the sub-modular tile design. The concept is to create a cubic structure that can be assembled from tiles with the same external dimensions. The concept for this design is discussed in the previous chapter. The outside dimensions of the tile are such that each can be identical, and assemble together to form a cube. This is achieved through tabs that protrude from the main body that allow the corresponding side of the adjoining tile to mate to it. The tiles are mechanically fastened to each other with M3 bolts that join the protruding tab to the side of the adjoining tile. The bolts are driven in from the leaf of one tile, in to the side of the body of the adjoining tile, and screwed in to a nut that is set in to a recessed location on the inside of the tile. The major dimensions of the tile design are depicted in figure 4.1. The design also includes four mounting holes for a PCB at each corner of the tile. These are dimensioned to use an M3 heat set insert.

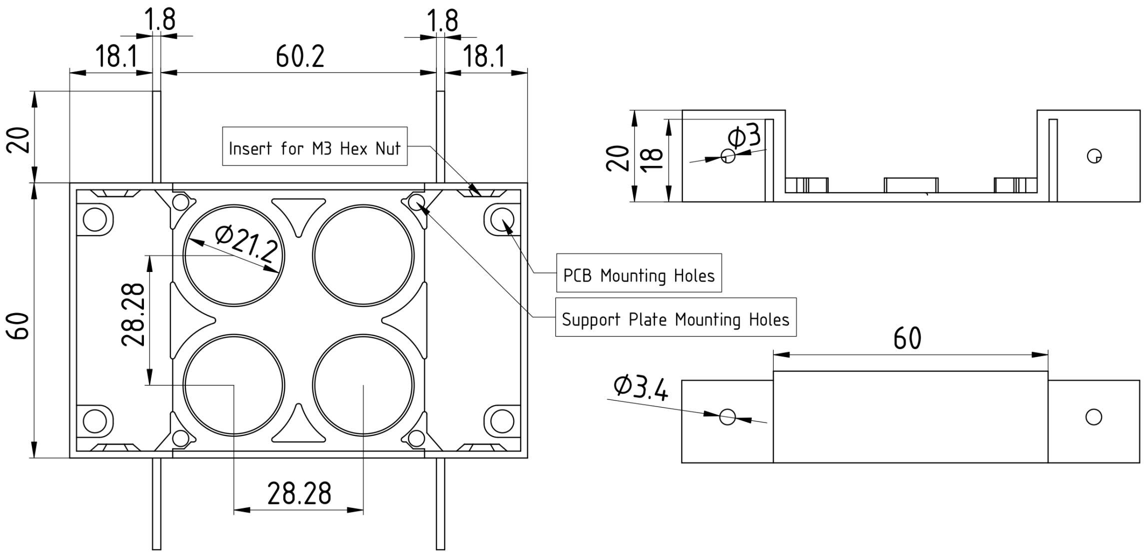


Figure 4.1: Mechanical drawing of the sub modular tile design showing all major dimensions.

4.3 Docking Connector

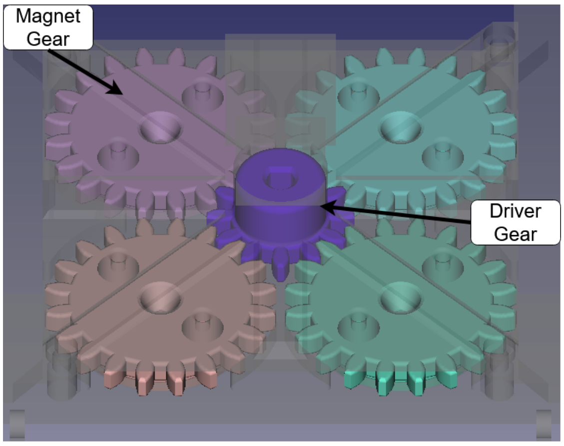
The concept of the docking connector design was discussed in the previous chapter. These design decisions laid the foundation for the connector design, but did not provide an implementable design. Where possible, this connector is constructed with 3D FDM printed parts. This will make this prototype easier to replicate for other researchers. The core of the design is the four rotating pairs of magnets. The connection mechanism has three states of operation.

- **Active Connect** - In this state the magnet poles of the face are aligned with the opposing magnet poles of the opposite face. This creates a net attract force between modules and holds them together.

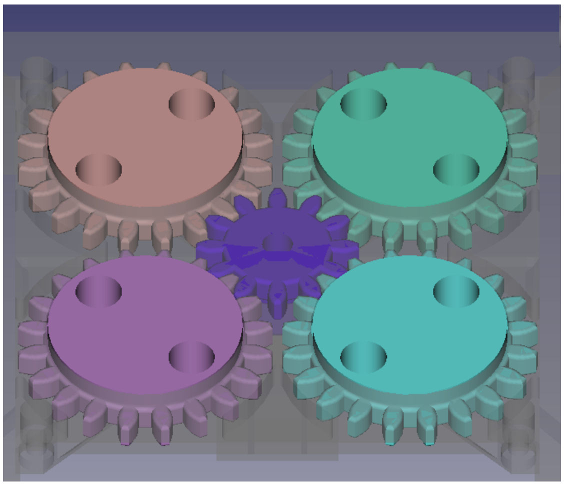
- **Active Disconnect** - In this state the magnet poles of the face are aligned with the same magnet poles of the opposite face, this creates a net repel force between modules and pushes them apart.

- **Passive** - In this state the magnet poles of the face are 90 degrees offset from the magnet poles of the opposing face, this creates a net zero force between the modules.

The connector is constructed from 7 components. There is the tile body, four magnet pair gears, a driver gear and a support plate on the inside of the tile. The driver gear of the connector, which directly couples to the micro metal motor has 12 teeth. This gear is centered through a clearance fit shaft in the support plate. The four magnet pair gears have



(a) Inside View



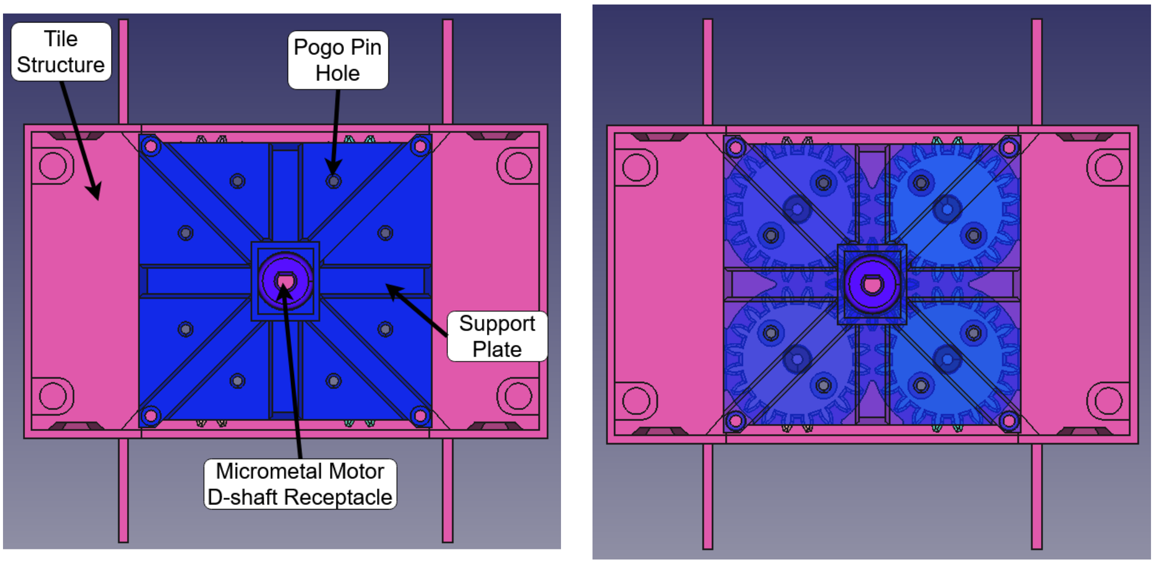
(b) OutsideView

Figure 4.2: Docking Connector Gear Arrangement

20 teeth each. The gears are held in place concentrically at the front and back sides. On the front side there is a clearance hole in the tile body, and at the back there is a conical shaft which mates to a corresponding shaft on the support plate. The magnets are pressed in to the appropriate holes in each magnet pair gear. The back side of the magnet is exposed so that a spring loaded contact on the tile PCB can make electrical contact. The support plate has holes that align with each of the magnets when in the active connect position. These parts are illustrated in figure 4.2. The support plate is bolted to the tile body with four M2 bolts in to heat set threaded inserts in the tile body. The support plate also holds the micro metal motor. An assembly of the connector can be seen in figure 4.3 and figure 4.4.

To keep the stack of components inside the tile as small as possible, the support plate could only be 2mm thick. Rigidity is important for the support plate, if it deforms, then the conical shafts that couple to the back of the magnet pair gears will become misaligned. To add strength, buttresses were added to the inside of the support plate. This adds considerable resistance to bending moments, while also allowing more airflow under the PCB. The distance from the back of the magnet to the PCB is constrained by the length of the spring loaded contacts.

The outside face of the connector is shown in figure 4.4. The face of the magnet gears is flush with the tile body, this is to allow modules to be slid over each other, for example when a module is being inserted into a pocket surrounded by other modules in a given configuration.



(a) Showing Support Plate

(b) Showing connector gearing

Figure 4.3: CAD design of docking connector (inside view)

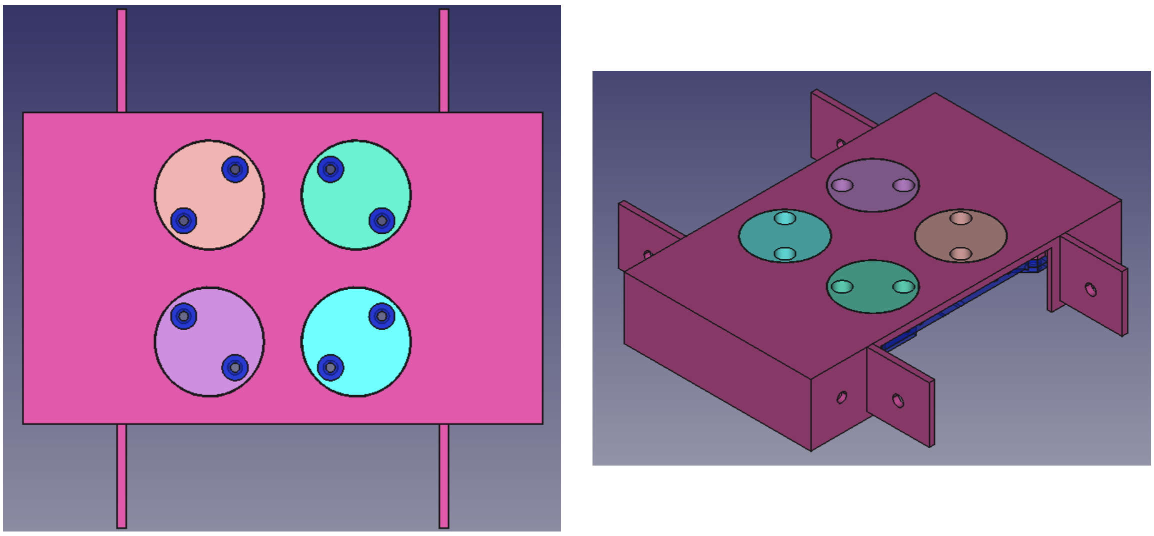


Figure 4.4: CAD design of docking connector (outside view)

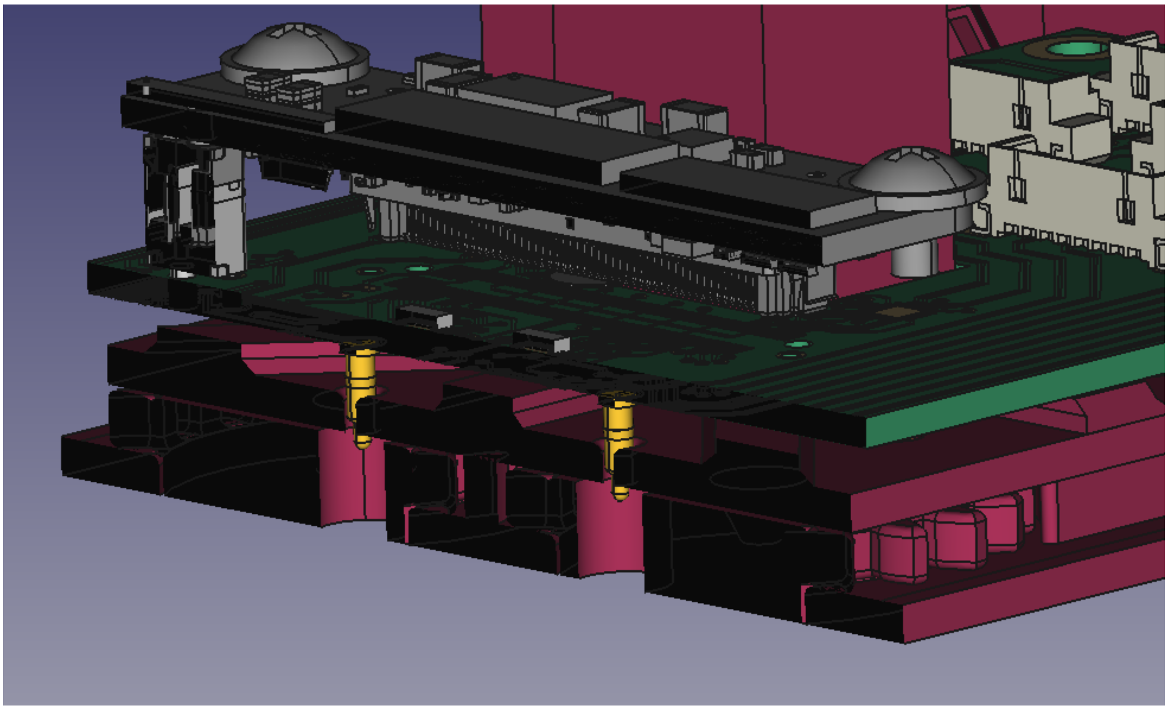


Figure 4.5: Section view of tile showing spring loaded contact connection to magnet holes

4.3.1 Inter-module Communication

The IEEE 802.3 100Base-TX Ethernet standard was used as a basis for the network hardware layer. 100Base-TX Ethernet uses two twisted pairs of copper conductors to operate full-duplex 100Mbps communication. One major benefit identified for modular robots is that some Ethernet physical layer ICs have the ability to detect and correct for crossover of pairs and crossover of conductors in pairs. This means that, when the conductors are arranged correctly, two modules can connect in any orientation and the Ethernet PHY IC will identify the connection configuration operate as normal. The Ethernet data pairs are connected between modules using the magnets that mechanically hold the modules as conductors. Each conductor is connected to two separate magnets, in the configuration shown in Figure 3.13 in the previous chapter. This configuration, where each conductor is routed through opposing sides of the magnet face, allows two tiles to connect in any orientation, and still maintain an active Ethernet channel. The stack up of the PCB with the spring loaded contacts through the support plate to the back of the magnets is shown in figure 4.5. When the magnet pair gears rotate or are in the passive state, the spring loaded contacts interface with the back of the gear. When the magnet pairs are in the active connect orientation, they interface with the back of the magnets.

4.3.2 Inter-tile Communication

Inside a module, each of the tiles need to have the ability to communicate with one another, share power, and route the Ethernet signal channels from the master tile to the slave tiles. CAN bus is used for the inter tile communication. This communication bus is responsible for low bitrate communication between tiles, used for tile connector actuation control and sensor streaming from sensor tiles. The system is architectured in such a way that if a higher bandwith than what CAN bus can provide is required, such as for a camera tile, the module connector Ethernet channel can be used to communicate with the master tile.

4.3.3 Master Tile PCB

The master tile PCB is the core of the electronics for each module. This tile is responsible for handling all network traffic processing and advanced computational capabilities. The module master tile is centered around a Xilinx Zynq 7020 SoC, which contains two arm processing cores and Artix-7 FPGA fabric. The FPGA portion of the device is utilised for networking protocol implementation and connecting to the 6 Ethernet physical layer transceivers. The Texas instruments DP83825 Ethernet PHY is used for this. Each transceiver then connects to the corresponding tile in the module, which contains the Ethernet transformer, and connection to the tile magnets. The PCB design is shown in figure 4.6.

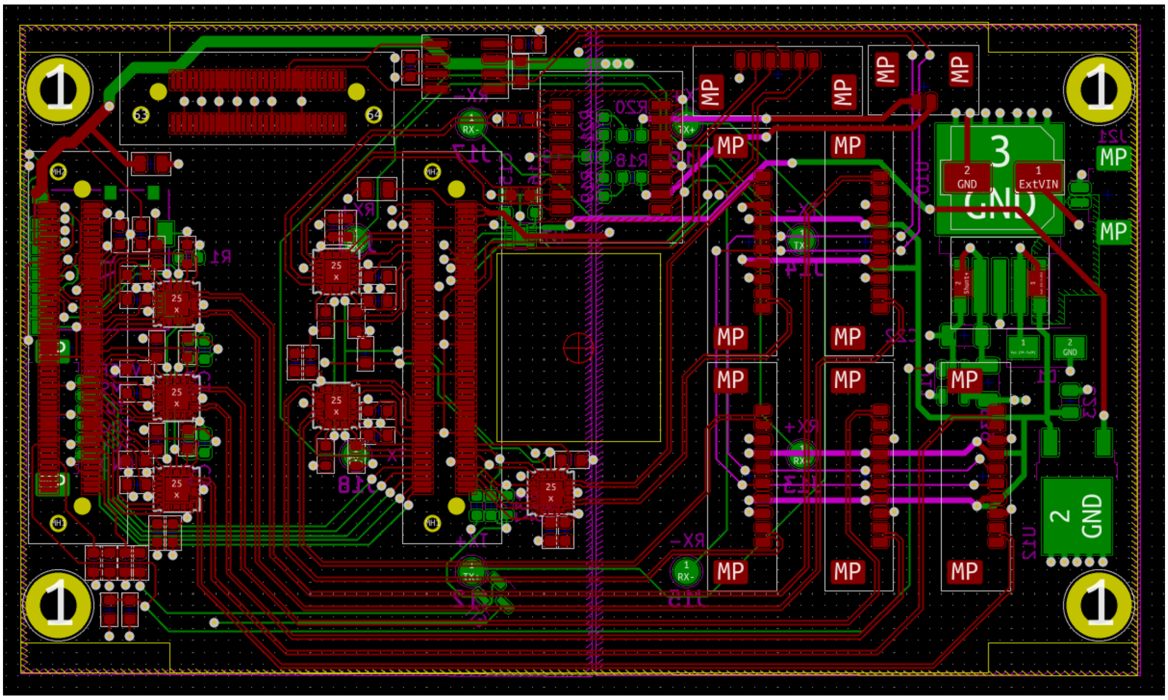


Figure 4.6: Master tile PCB design

The PCB has been designed around a Trenz electronic TE0720 System on Module (SoM)[95]. The SoM is a 4cm x 5cm PCB that contains a Zynq 7020 System on Chip (SoC), 1GB DDR3 RAM and 8Gb e.MMC non-volatile memory. It is designed to connect to a custom carrier board with board to board connectors for application specific uses. The Trenz SoM connects to the module board through three Samtec Razor Beam[96] board to board connectors. This allows I/O from the FPGA portion of the Zynq device to be routed to the various peripherals on the board. The PCB is composed of four layers. The stackup is Signal-GND-Power-Signal. Some signals had to be routed on the internal layers around the tile to tile connectors to give enough clearance for the MDI interfaces routed from the Ethernet PHYs

The six Texas instruments DP83825 Ethernet PHYs are connected to the FPGA through the RMII interface standard. RMII (Reduced Media Independent Interface) is an interface for connecting Ethernet PHYs with Ethernet media access controller (MAC) hardware. This is usually an embedded peripheral in a microcontroller or microprocessor, an Ethernet switch ASIC, or an FPGA. The RMII standard was developed to reduce the pin count required for systems that were previously built on the MII (Media Independent Interface) standard that was used previously. The MII Standard is composed of 16 pins for data transfer, whereas the RMII standard is composed of 7 (6 data and a clock). The RMII interface supports both 10Mb/s and 100Mb/s data rates. Table 4.1 contains the pin names and descriptions of each. The TX and RX data pins are formed of two signal dibits. The synchronous clock operates at 50MHz.

Table 4.1: Pin Descriptions of the reduced media independent interface (RMII).

Signal Name	Direction with respect to the PHY	Description
REF_CLK	Input	Synchronous clock reference for data interface
CRS_DV	Output	Carrier Sense/Receive Data Valid
RXD[1:0]	Output	Receive data
TX_EN	Input	Transmit Enable
TXD[1:0]	Input	Transmit Data

The clock for each PHY is sourced from a PLL on the Zynq device. The DP83825 PHY chips were chosen primarily for their small form factor of 3x3mm, the smallest footprint 100Base-TX PHY on the market. 5 of the six PHYs are situated underneath where the trenz SoM is located. Signal integrity is an important factor to consider when designing Ethernet systems, and the easiest way to ensure the best signal integrity is to keep the signal traces as short as possible. It was not possible to fit the last PHY underneath the SoM, as space had to be available for the MDI interface signals to be routed to the intra tile connectors. Routing signals from one side of the board to another proved challenging as there is limited board space between the two sides due to the hole in the middle of the board where the micro metal motor that is used for connector actuation is located.

Located at the top middle section of the board is the Ethernet transformer for the master tile and the CAN transceiver for communication with other tiles. On the right hand side of the board are the 5 intra module tile connectors that route the MDI interface, CAN bus and power to the other tiles. There is also the header for connecting to the tile connector motor, and on the back side there is a 3.3V switching regulator and a 6V regulator for the motor driver.

For sensing the current position of the connector mechanism there are two systems in place. A quadrature encoder is affixed to the back shaft of the micrometal motor, and there is a hall effect sensor adjacent to one of the magnet positions. This gives the ability to perform a homing sequence with the connector using the magnetic field of one of the magnet pairs.

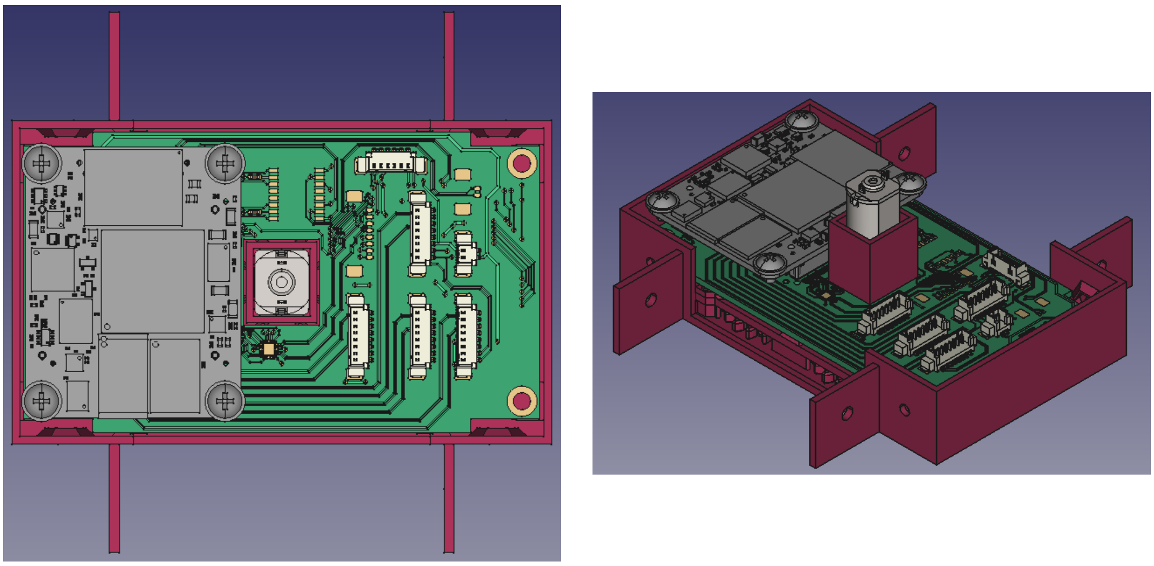


Figure 4.7: CAD render of Master tile with PCB and Trenz Zynq 7020 system on module

The render shown in figure 4.7 shows an assembly view of the master tile. This contains the 3D printed parts of the tile and connector mechanism, the master tile PCB in situ, the Trenz TE0720 SoM, and the micro metal motor. This view shows how compact the whole design had to be to fit in to a 10x10 tile. There is 0.5mm of space between the SoM module and adjacent plastic components. There is also a cut out introduced on the side of the tile, this is to allow extra air volume above the SoM to aid with thermal management.

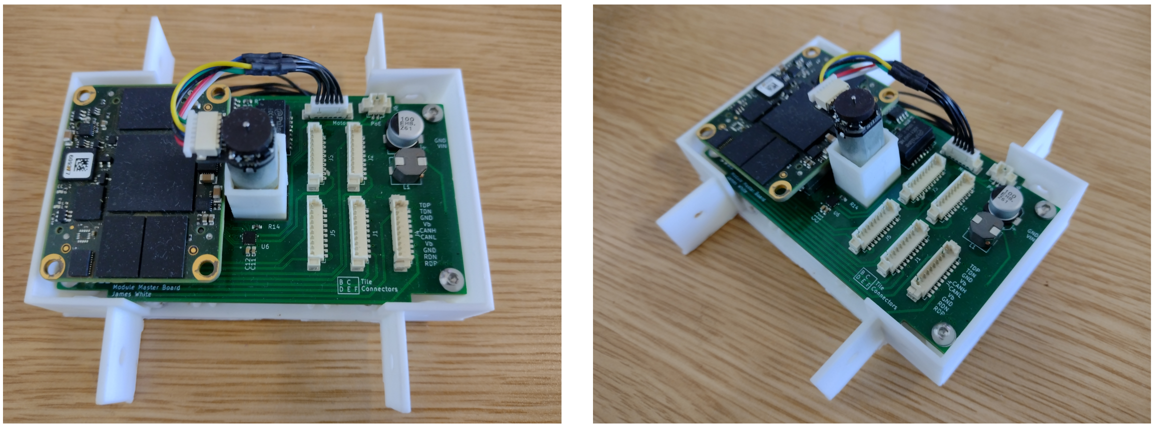


Figure 4.8: Assembled Master Tile

Figure 4.8 shows the manufactured and assembled master tile. The encoder can be seen on the top of the micro metal motor. The PCB was manufactured by Eurocircuits GmbH and populated by the author using SMD reflow equipment at the University of York. The 3D printed plastic parts were manufactured using an Intamsys Funmat HT FDM printer at the University of York with ESun PLA filament.

4.3.4 Slave Tile PCB

The slave tile PCB, shown in figure 4.9 is responsible for control of the connector motor on the tile it is connected to, routing of the MDI signals from the master tile through the Ethernet magnetics to the magnets on the connector and providing capability to add extra functionality to a slave tile through GPIO breakout.

It is composed of two copper layers. It contains an STM32F334 microcontroller which is responsible for handling CAN bus communication with the master tile and controlling the motor and GPIO breakout. This was the smallest footprint microcontroller with a CAN peripheral that could be used. The board also contains a 3.3V voltage regulator, CAN transceiver, 6V regulator for the motor, and DRV8837 Motor controller. The GPIO breakout is via a 2.54mm pitch male header. This will allow small boards to be made that can stack onto that connector. Uses for this can include sensor controller electronics, servo control and other extensions. The Ethernet signal traces are length matched from the output side of the Ethernet transformer to each of the pogo pins.

Figure 4.10 shows a CAD render of a slave tile containing all of the plastic components, the slave PCB, as well as the micro metal motor used for actuating the tile connector. The

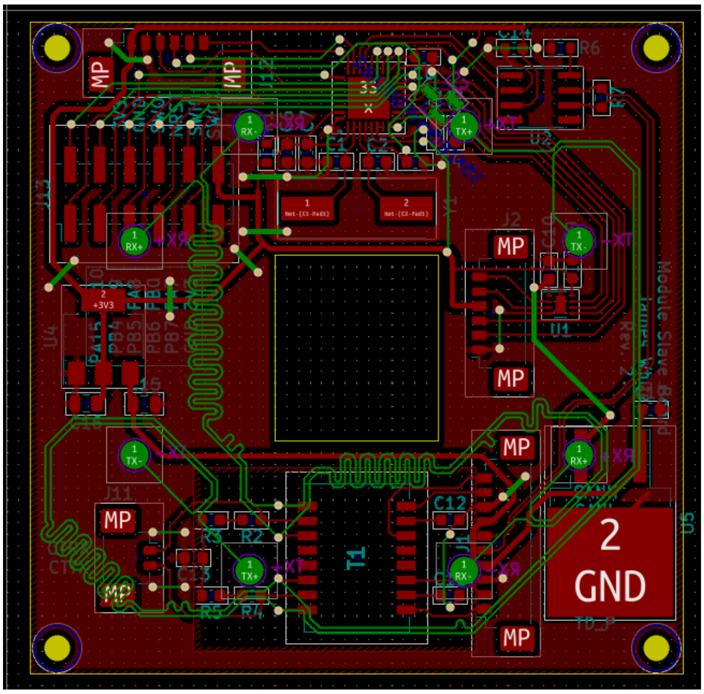


Figure 4.9: Slave Tile PCB Design

slave PCB mounts to the tile through the mounting points for the support plate. The plastic components are an identical design to that of the master tile, and no modifications to the design are necessary to be used as a slave tile rather than a master tile.

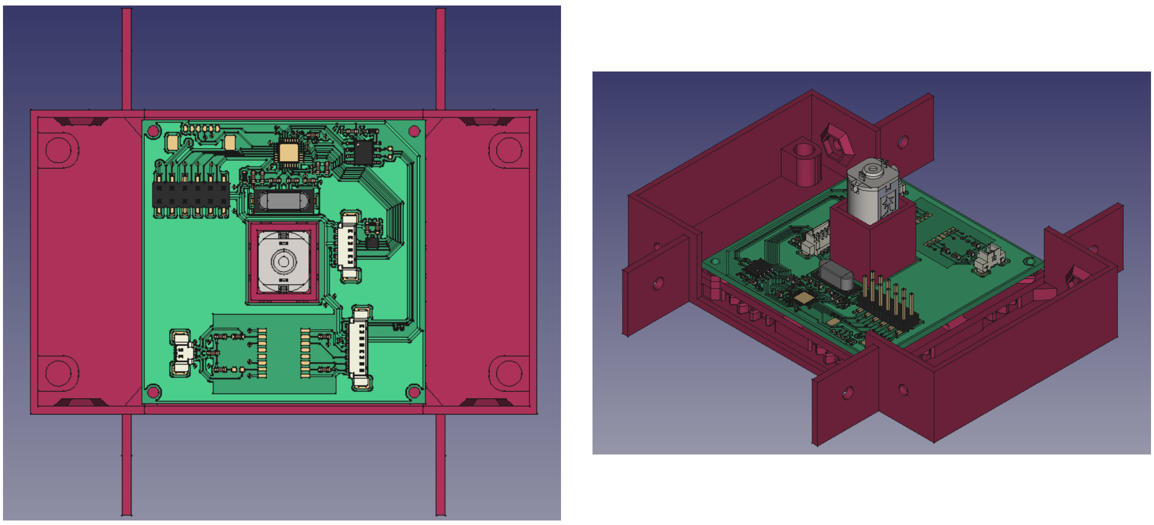


Figure 4.10: CAD render of slave tile with PCB and motor

Figure 4.11 shows the slave tile assembled. It contains all of the 3D printed plastic parts, the slave tile PCB, and the micro metal motor used for actuating the connection mechanism.

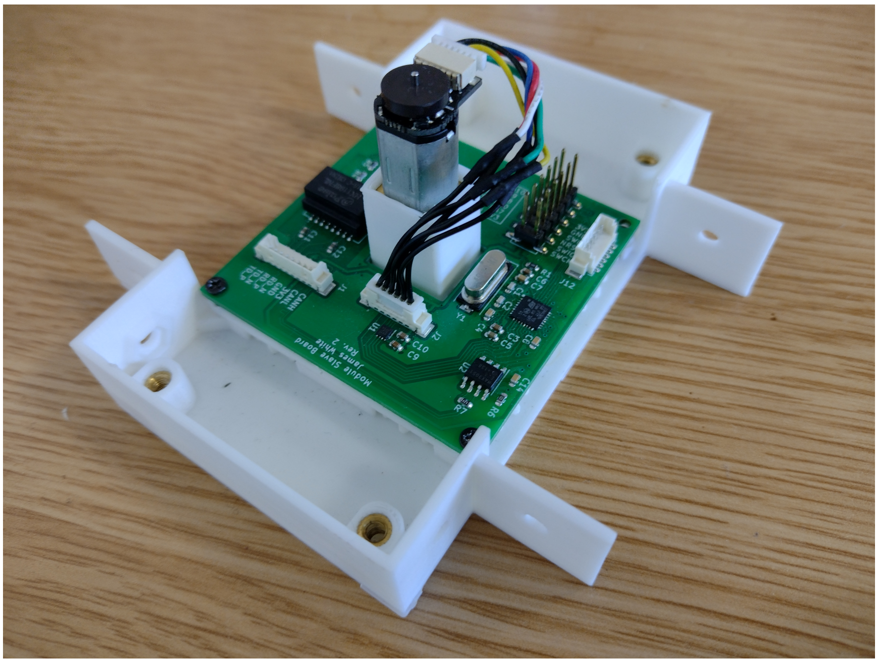


Figure 4.11: Assembled slave tile

4.3.5 Static Module

A static module is the main building block of a Mo* modular robot configuration. It is static in that it has no degrees of freedom other than the connector mechanisms embedded in each of the six tiles. This module contains one master tile and 5 slave tiles. A static module has the function of acting as a physical building block of the system, as well as a node in the network of modules and is responsible for forwarding packets between modules. As the static module contains a Zynq device, it can also be used for computation capabilities. One scenario where this would be useful is if there was a configuration containing one or more sensor modules, and some sensor fusion algorithm needed to be applied. Each of the sensor modules could stream its data to the static module, and it could process the data in a SLAM algorithm, for example.

Each of the tiles in the module are connected through a 10 pin connector that routes power, CAN and Ethernet between tiles. There is a shared CAN bus throughout the module. The CAN bus is composed of two conductors CANH and CANL. These are terminated with resistors at each end of the bus. As all of the slave PCBs are identical, the termination resistors are not populated on four of the slave tiles. The master tile and one of the slave tiles contain the termination resistors. The architecture of the CAN bus is shown in figure 4.12.

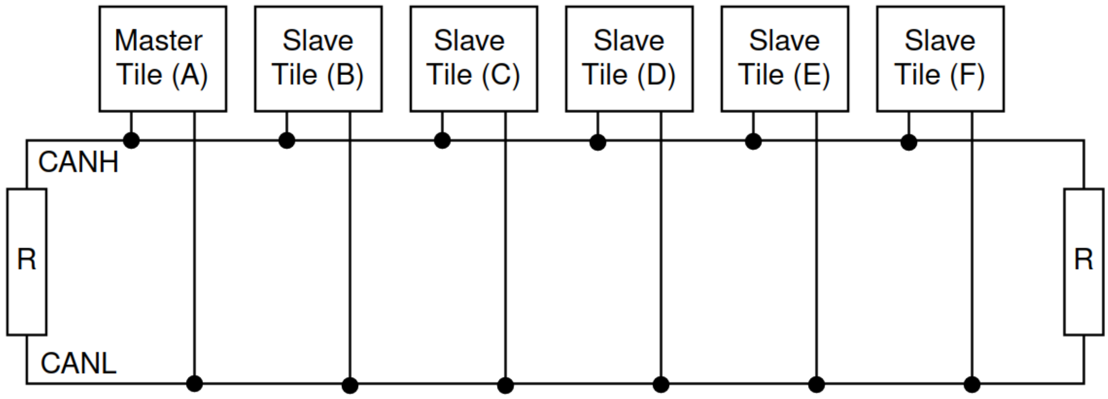


Figure 4.12: Architecture of the CAN bus in a module

The inter tile connection is achieved via a 10 pin Molex picoblade connector and cable. The pinout of this connection is:

1. TD_P
2. TD_N
3. GND
4. VBus
5. CANH
6. CANL
7. VBus
8. GND
9. RD_P
10. RD_N

The Ethernet signals are separated from the CAN signals by a power and ground line on each side. This is to minimise crosstalk between them.

Figure 4.13 shows a CAD render of three tiles assembled together. This shows how tiles and the electronics in them physically interact. The very small spacing betewen the Trenz SoM and the adjacent tile's slave board can be seen here. This also shows the spacing between the micro metal motors for each tile. A CAD render of 5 Tiles assembled is shown in figure 4.14. Assembling a module is not trivial, as there is little room for the inter tile cables and they have to be kept clear of the motors. In a future design, a board to board connector setup could be implemented as this would greatly ease the assembly process.

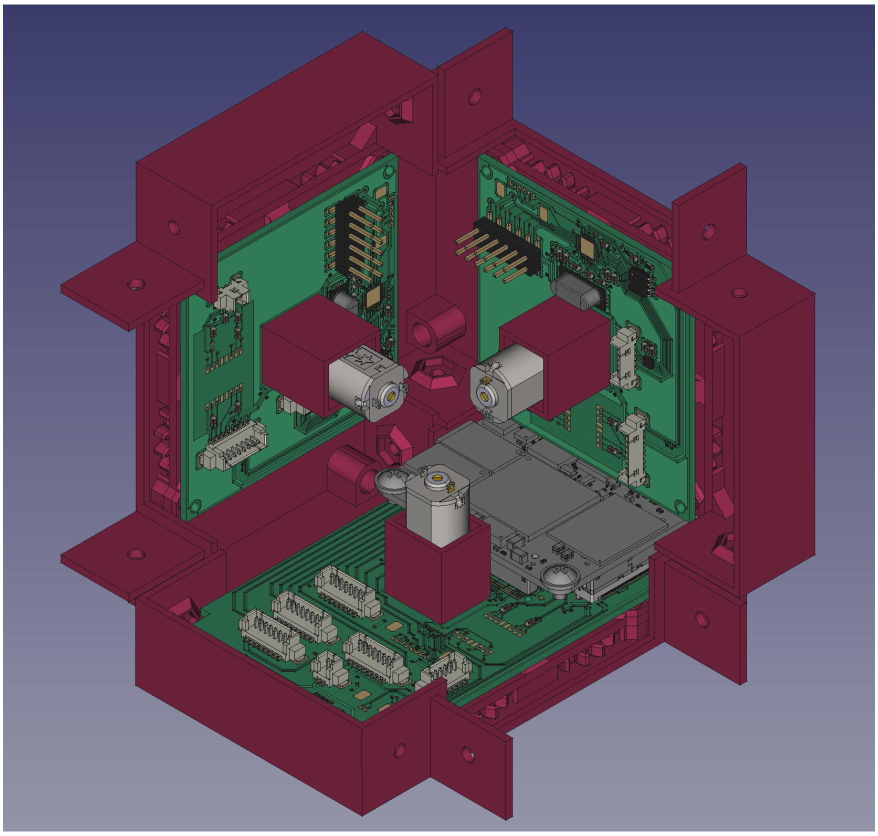


Figure 4.13: CAD render of 3 tiles

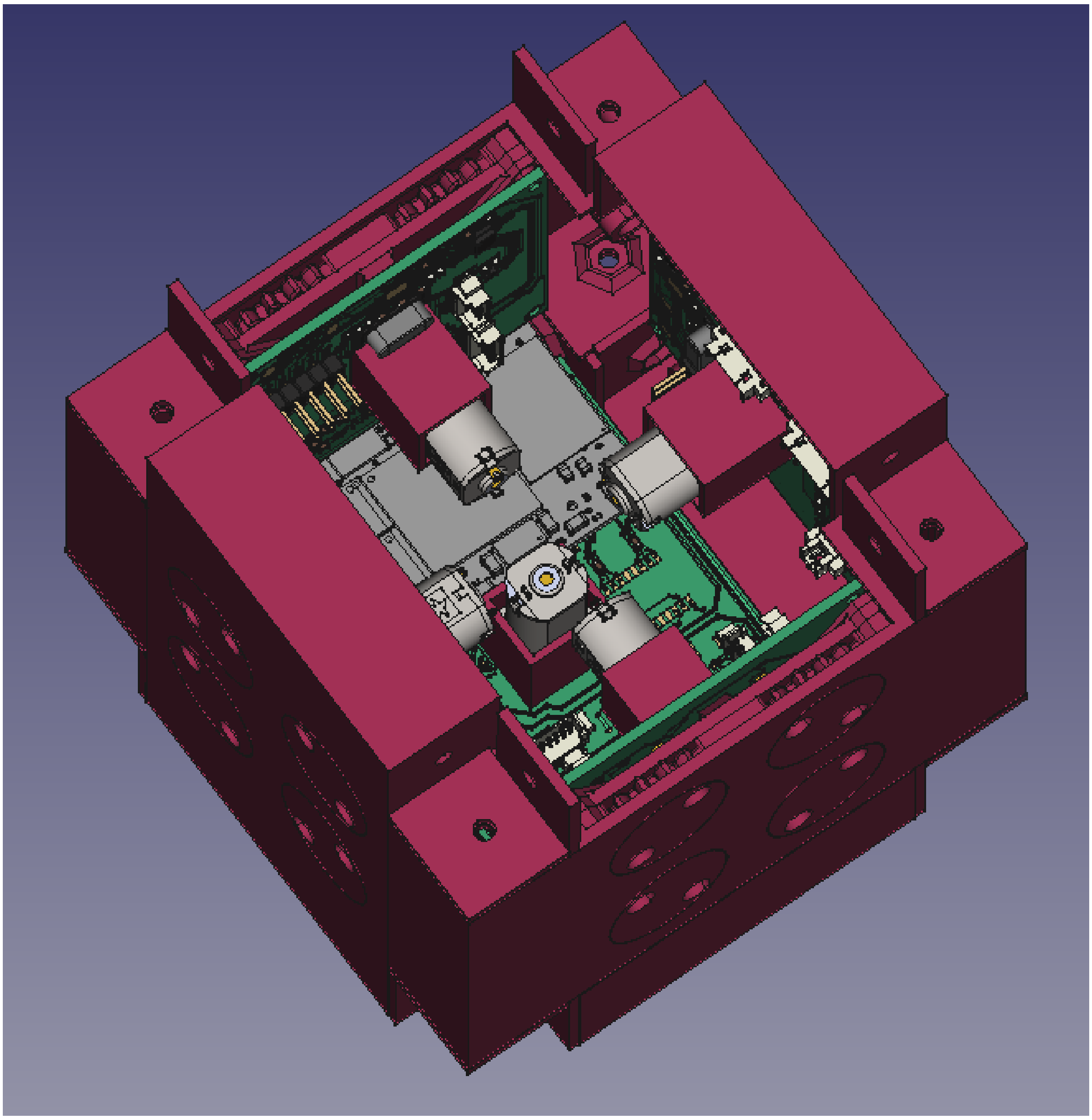


Figure 4.14: CAD render of 5 tiles showing spacing between connector motors

4.4 Embedded Software

Software Architecture overview

Each static module contains five STM32 F334 Microcontrollers, one on each slave tile, and a Zynq SoC containing two Arm Cortex A9 processing cores and Artix-7 FPGA fabric, on the master tile. The master tile is responsible for handling all network processing, the motor control for that tile, and CAN bus communication. The slave tile is responsible for handling CAN communication, motor control of that tile, and GPIO to extension boards.

4.4.1 Master Tile

Ethernet PHY Interface

The RMII interface standard is defined in [97]. This document specifies both the hardware and software components of the specification. This was used to create an FPGA IP core for interfacing with the PHY devices over the RMII interface. The RMII interface uses a frame format to send data to the PHY. The structure of this frame is <interframe><preamble><sfd><data><efd>. In the inter frame state RXD/TXD[1:0] is "00". The preamble state is used for ensuring that the clock is synchronous with the data between the MAC and the PHY. This consists of RXD/TXD[1:0] = "01" and is seven octets long. The sfd, or start of frame delimiter is a byte of "01010111" which signals the start of the data transmission. Data is then transmitted, and can be any length. The EFD is signalled when TX_EN is de-asserted. Figure 4.15 shows the bit level data flow sequence for data reception on the RMII interface. Figure 4.16 shows the bit level data flow for data transmission on the RMII interface. State machines were implemented for handling transmission

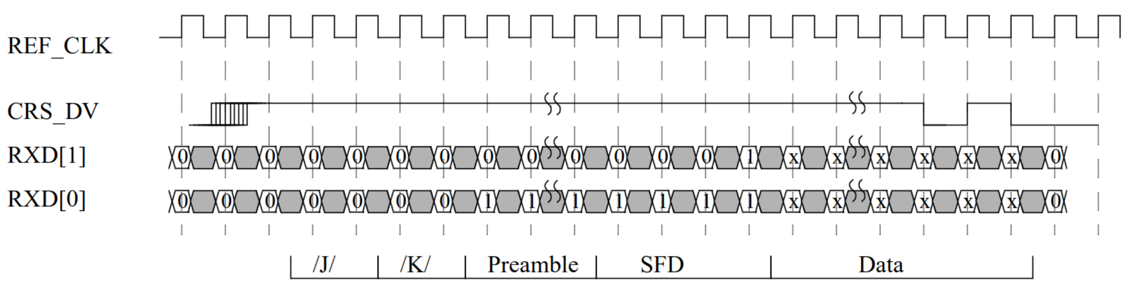


Figure 4.15: Bit transitions of frame reception on the RMII interface.[97]

and reception on the RMII interface. There are 4 main state machines operating. One for generating the di-bit stream and handling the TX control signal on the RMII interface from

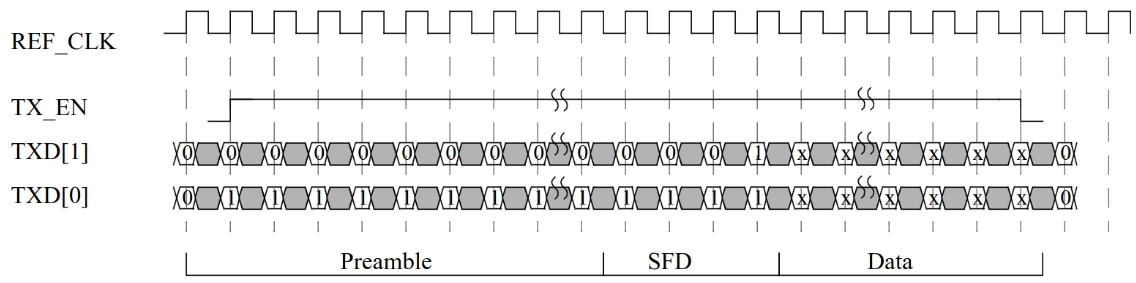


Figure 4.16: Bit transitions of frame transmission on the RMII interface.[97]

byte-wise data, one for receiving the di-bit stream on the RMII interface and converting it to byte-wise data, one for frame transmission, and one for frame reception. The state transition diagram for the frame transmission state machine is shown in figure 4.17. The

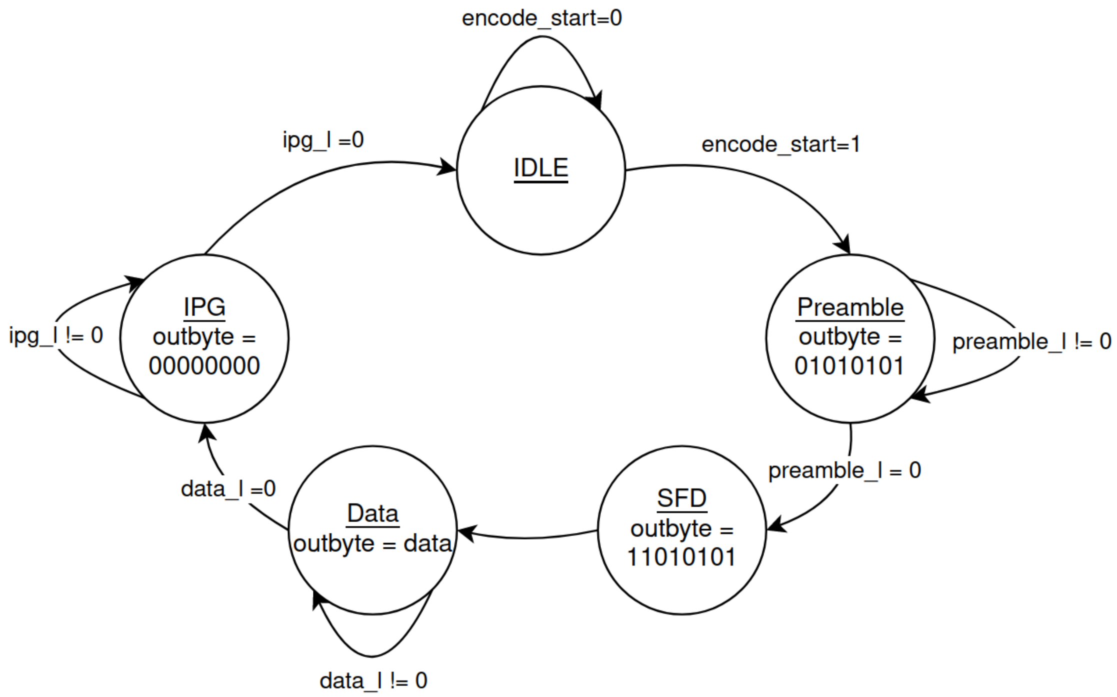


Figure 4.17: State transition diagram for the state machine that handles RMII frame transmission.

state transition diagram for the frame reception state machine is shown in figure 4.18 The MDIO (Management data input/output) interface is used for accessing register space of the Ethernet PHYs. It is defined in the IEEE 802.3 standard. It is a serial bus consisting of a clock and one data signal line. The data line is bi-drectional. The MDIO standard defines a packet structure to use to communicate with PHYs. The packet structure is <Pream-

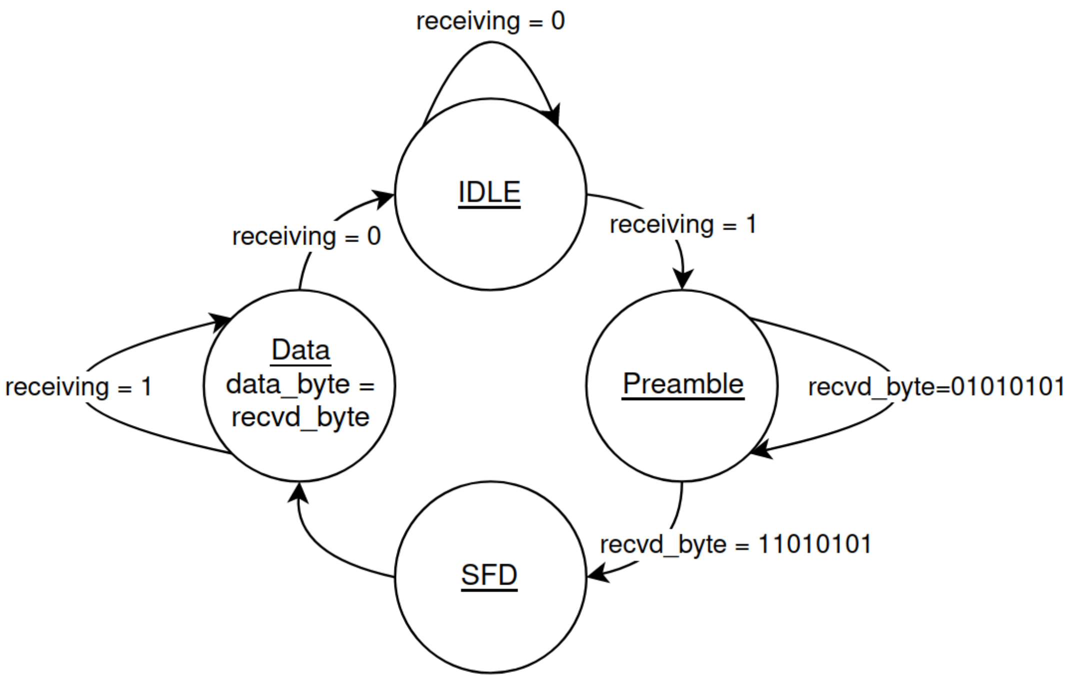


Figure 4.18: State transition diagram for the state machine that handles RMII frame reception.

ble><ST><OP><PHYADDR><REGADDR><TA><Data>. The preamble is 32 bits long, consisting of repeating '1'. ST is the start field that is two bits long and always "01". The OP field is the OP code which denotes whether the packet is a read ("10") or write ("01") operation. PHYADDR is the address of the PHY to be accessed. The standard allows up to 32 phys to share the same mdio bus. REGADDR denotes the register to be read from or written to. TA is a turnaround field. When data is being written to the PHY, the MAC writes "10" to this line. When data is being read, the MAC releases the data line for reception. DATA is a 16 bit data field for reading from or writing to the register space at the address specified earlier in the packet. In total a packet on the MDIO line is 64 bits long. The MDIO clock can be user defined, based on the chip being used. The DP83825 chip used in this system allows a maximum clock rate of 25MHz on the MDIO bus. The DP83825 PHY only supports up to four PHYs sharing an MDIO bus, so two seperate MDIO buses had to be used in this system.

Motor Interface

The motor that actuates the master tile is driven by the DRV8837 Low-voltage H-bridge driver from Texas Instruments. This has two motor control inputs and a sleep input. There is a quadrature encoder attached to the motor which has two outputs. The absolute position of the tile connector is obtained by a DRV5055 hall effect sensor that provides an analog output proportional to the observed magnetic field strength at a normal to the top of the device. All of the digital signals are electrically routed to the FPGA portion of the Zynq device and then internally routed to the EMIO peripheral of the Arm microprocessors. This allows direct control of the motor and reading of the quadrature encoder from the processor portion of the Zynq device. The analog output from the hall effect sensor is routed to an XADC peripheral on the Zynq device, and can be monitored from the ARM processors through the AXI interface. At power on, if the connector is not connecting to another module at present, the tile controller initiates a homing sequence where the connector rotates 1 full revolution counted by the number of steps from the encoder. The analog input from the hall effect sensor is also sampled at this stage, and where a positive peak forms is taken as the home position, as this is where the south facing side of a magnet is passing over the sensor. The connector then goes to its home position.

CAN interface

There is a CAN interface for communicating with other tiles in the module. To implement this, the embedded CAN peripheral on the Zynq device is used. This interfaces with the ARM processing cores on the device directly. Xilinx provides a library for interfacing with the CAN peripheral. This library provides functions to set up the CAN device, pass data to a TX buffer, and receive data from an RX buffer. The CAN interface is primarily used to allow the master tile to control the docking connectors on each of the slave tiles. The master tile sends a message to the slave

4.4.2 Slave Tile

The embedded software on the slave tile is primarily responsible for communicating with the master tile via the CAN bus and controlling the motor that actuates the tile docking connector. There is a CAN peripheral built in to the microcontroller. The device supports a bit rate of up to 1Mbit/s. Motor control is achieved using the same devices as on the master tile. At start up the microcontroller homes the connector, sweeping the connector through one full rotation while keeping track of encoder steps and sampling the hall effect sensor that is connected to one of the analog inputs on the device. After this the home position of the connector can be determined and the controller moves the connector to that position. After this, the controller sits idle waiting for a CAN message from the master tile to initiate any other actuations that are required. The slave tile board also includes a 12 pin GPIO extension connector. This has 10 GPIO connections and two power connections. Combinations of these pins can be used as digital IO, Analog IO or SPI,UART or I2C communication. This provides flexibility in designing extension boards for the slave tile.

4.5 Mo* Network Protocol

This section outlines the work done to implement a novel network protocol for the Mo* modular robot platform.

The hardware layer of the networking system is built around IEEE802.3 compliant Ethernet PHY transceivers. As such it is required to use the standard Ethernet frame format as the base data unit of the system. An Ethernet frame consists of a preamble that is used to ensure that the two connected PHYs at either end of the MDI link are synchronised. This is a stream of 7 bytes of "10101010". Following this there is a start of frame delimiter byte of "10101011". After this there is a field containing the destination MAC address followed by the source MAC address. These are each 6 bytes long. The MAC address is a unique identifier of the Ethernet MAC. After this there is a field containing the length of the frame, specified in the standard as a maximum of 1500 bytes for the data field. Finally there is a cyclic redundancy check field which is used to confirm that a packet has been successfully received. The structure of the packet is outlined in table 4.2.

Table 4.2: Format of an IEEE802.3 compliant Ethernet frame.

Bytes	1-7	1	6	6	2	46-1500	4
Field	Preamble	SFD	Dest MAC	Source MAC	Length	Data	CRC

4.5.1 Packet Format

On top of these Ethernet frames, the Mo* protocol packets are used as the packet for data transmission around the network. The aim of this protocol was to implement a system that is as simple as possible. Keeping the packet header to a minimum increases the rate at which usable data can be propagated around the network. It is the hope that this protocol could be implemented on other hardware mediums in modular systems. As such it is required to have another form of ID in the protocol header. If the protocol were only to be used on Ethernet hardware layers, then the MAC address could be used as the only identifier for modules, as this can be set in the packet construction logic in the FPGA. At current, the maximum number of modules expected to be in a network is less than 125, which would be a solid three dimensional structure of 5x5x5 modules or larger structures that are not completely filled, such as a rover configuration. As such a one byte field is used for each module ID.

The packet begins with the destination ID, then the source ID. IDs are assigned during the topology discovery phase of the protocol. After this there is a data class field. This is used to allow efficient processing of packets. The relevant protocol state machine is assigned to process a packet based on this data class field. Following this there is a length field, which is equal to the length of the packet including the header. Finally the header contains a checksum that is uses to check the integrity of the packet. The data in the packet can be a minimum of eight bytes long and a maximum of 65529 bytes long. This is equal to $2^{16} - headerlength$. If these packets are larger than the maximum Ethernet frame size, then they are fragmented across multiple Ethernet frames and reassembled on the receiver side.

Table 4.3: Mo* Protocol Format

Bytes	1	1	1	2	2	8-65529
Field	Dest ModID	Source ModID	Data Class	Length	Checksum	Data

4.5.2 Summary

This chapter has outlined the physical implementation of hardware and software for the Mo* modular robot platform. The design concepts from the previous chapter were used as a specification for the implementation design. A 3D printed tile and docking connector design that can be used for both master and slave tiles has been implemented. A PCB implementation for the master tile which includes a Zynq 7020 based Trenz TE0720 SoM, 6 Ethernet PHYs, motor controller and CAN communication is discussed. A slave tile PCB implementation has also been created which consists of an STM32F334 microcontroller, motor control, CAN communication capabilities and a general purpose extension board header. The embedded software implementation on these devices has been discussed. Finally the network protocol implementation that has been used to allow modules to communicate with eachother through the Ethernet hardware layer implementation is detailed.

Chapter 5

Verification

This chapter describes the verification of the Mo* platform in simulation and hardware. A discussion on why a new simulator was created is presented. The architecture of the simulation platform is presented. The capabilities and limitations of the simulator are detailed. Physicial implementation verification is undertaken to evaluate the performance of the Mo* modular robot platform. The mechanical properties of the connection mechanism are evaluated. Following this, The experiments undertaken to evaluate the electrical characteristics of the docking connector mechanism are outlined.

5.1 Simulation

To aid development of modular robot platforms, simulation tools can be utilised to test varying aspects of the system before physical prototypes are manufactured. The functionality of simulation tools varies greatly. Simulation tools can be used to simulate one particular aspect of a system, such as how a kinematic machine interacts with its intended environment. In simulating robotic systems, it is useful to be able to simulate as many aspects of the system with as much fidelity as is practical. With many simulation environments, certain scripting languages and routine calls have to be used which makes porting code to a hardware platform difficult. In this work a simulation environment has been created that simulates the physical aspects of a modular robot system, as well as the control code and networking between modules in a representative environment

5.2 Why create another simulation environment?

There are many robotic simulation environments available to researchers [98–101]. The limitations of these systems is that none provide the capability to simultaneously and accurately model the physical as well as networking and control aspects of a modular system. With popular simulation platforms such as Gazebo added [102], tradeoffs in the simulation of the networking and control have to be made. Gucwa et al undertook the task of creating a simulation tailored for modular robots with RoboSim added [75]. For accurate full stack simulations of modular robots, there needs to be a holistic simulation environment. The gold standard for simulation of modular robots would be integrating a flexible physical simulation environment such as unity, with a network simulation platform such as NS3 added [103], and potentially even extending to integrate processor emulation tools such as QEMU added [104] so that code written in simulation could be directly ported to hardware. A simulation environment such as this would greatly speed up the development of modular robotics platforms.

5.3 Simulation Overview

To enable development of networking protocols, reconfiguration algorithms, and task planning in the Mo* modular robotic system, a simulation environment has been created to model the relevant aspects of the system. The simulation environment contains two main components. The physical simulation is performed using the Unity engine, and a custom POSIX compliant network simulator has been created to model the network interconnections between modules. Unity was chosen as the physical simulation engine as it is has a very flexible implementation. All aspects of the unity simulation environment are customisable through a common c# interface. Unity uses PhysX for its physics engine. The network simulator is time synchronised from the Unity simulator. Arbitrary module configurations can be created in the simulator. Currently a model of a kinematically static module and a wheel module exists. With this, modular rovers can be created in simulation, like that shown in Figure 5.1.

The high level architecture of the simulator is shown in figure 5.2. The simulator is broadly split in to two parts; the backend simulator, and the unity simulator. The Unity simulator uses The Unity engine added [105] to simulate all of the physical aspects of the system including the connecting forces of the docking connectors. The backend simulator handles all of the electrical and software aspects of the modules including the network model. This has been created from scratch in C++. Attempts were made to integrate network

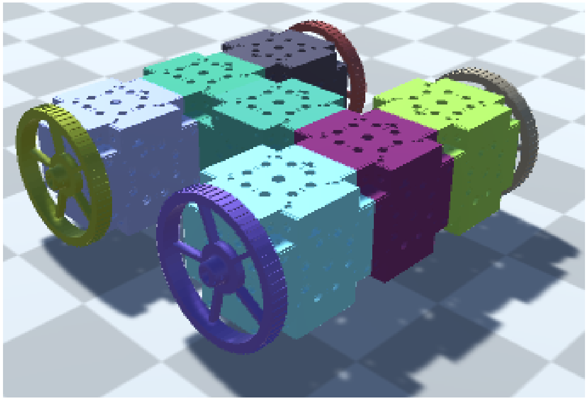


Figure 5.1: A modular rover configuration in simulation

simulators such as NS3 in to the Mo* simulation environment, but the complexities of integrating the physical simulator with existing network simulators proved more challenging than creating a network simulator designed to integrate effectively. The two parts interface with each other using inter process communication via the sockets API added [106] in linux.

The aim of the simulator was to create a flexible simulation platform for modular robotics in which it is relatively simple to alter components of the simulator for different scenarios. The back end simulator is split in to various components for the different parts of the system. There is a unity sim interface module that handles the passing of data between the backend simulator and the unity physics simulator through the sockets API. This interfaces with a simulation manager module that is responsible for managing data flow between and synchronising the Ethernet channel simulator and the module manager. The module manager is responsible for managing the data flow between the various parts of each of the module instances. In an attempt to make the simulator as versatile as possible, each module instance itself was split in to a PHY emulator, FPGA functions emulator and Module code section. The goal of this was to make it possible to write code for the different parts of a module between the application code and the FPGA, and test this with different PHY device implementations. In hindsight, this was too optimistic an undertaking. A simpler architecture would have allowed much more time for experiments to be run. It is hoped that the current implementation can be built upon to generate a unified simulation environment for modular robots.

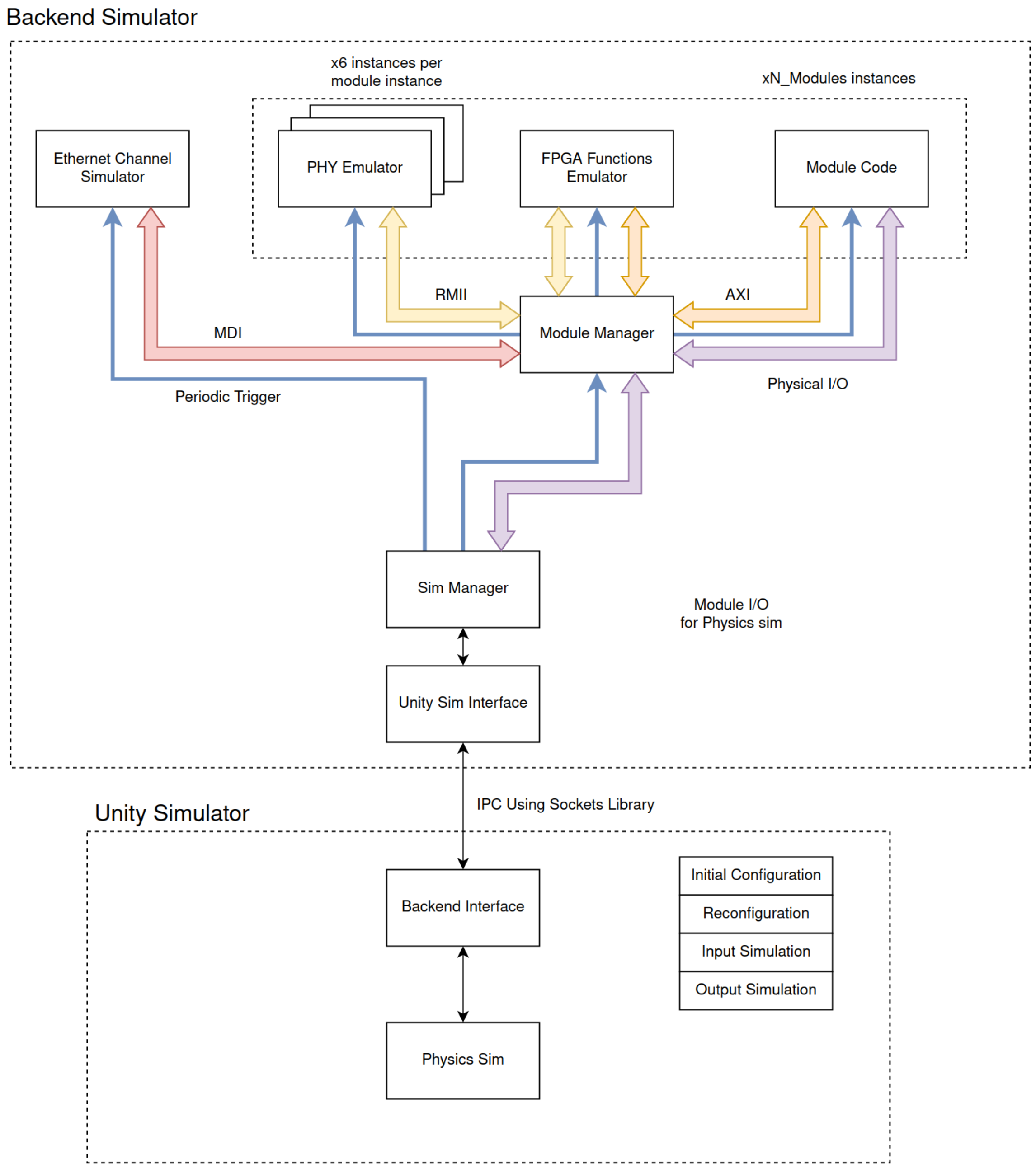


Figure 5.2: Architecture of the Mo* simulation platform.

5.4 Simulation Experiments

Experiments have been run to test the models of the network side of the simulator. Experiments were run on a 3x3x3 module strictly orthogonal network of kinematically static modules. A broadcast packet of varying size is sent from one of the corner modules, and the time taken for that packet to reach all other modules in the system is recorded for each packet length. This experiment is repeated with packet payload sizes from 1-5 bytes. The results from this experiment is shown in Figure 5.3. These results show the latency of the packet propagation increasing with payload size as expected. A render of the network of modules used in this simulation is shown in Figure 5.4. Once full integration of a module's mechanical and electronic components have been created, this experiment can be repeated in hardware to test the reality gap of the simulator, and tune the parameters of the models used.

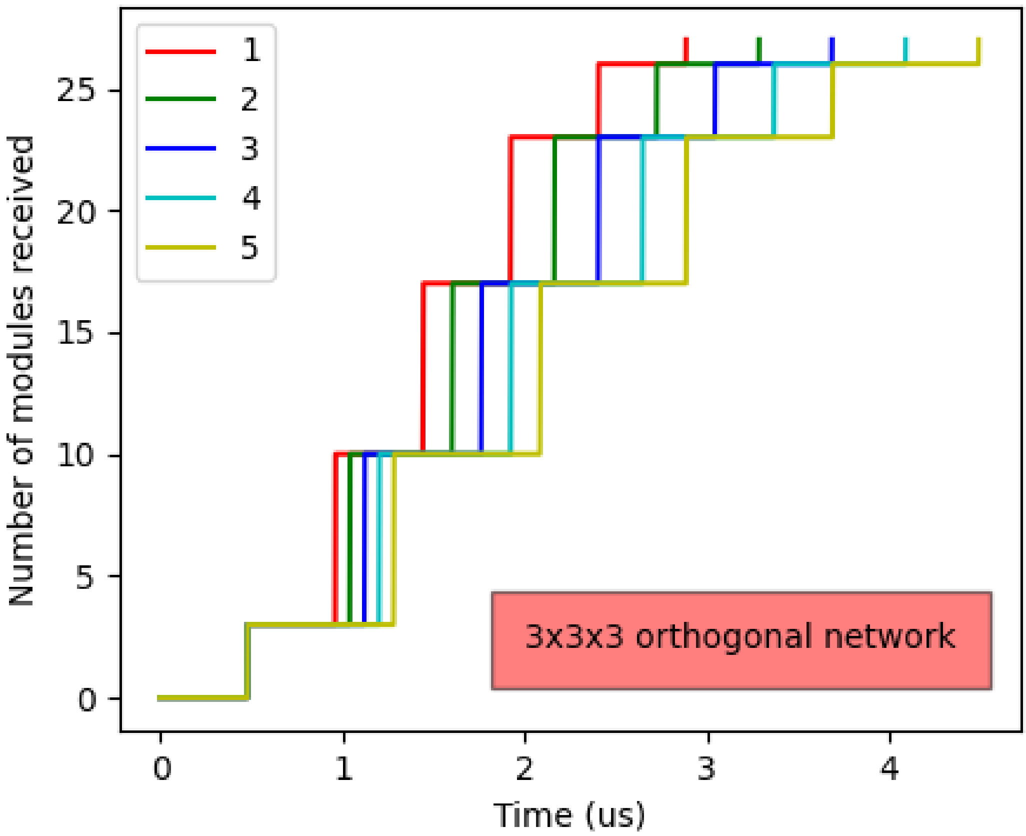


Figure 5.3: Simulation results of propagation delay of a broadcast packet in a strictly orthogonal network for varying packet payload sizes (bytes).

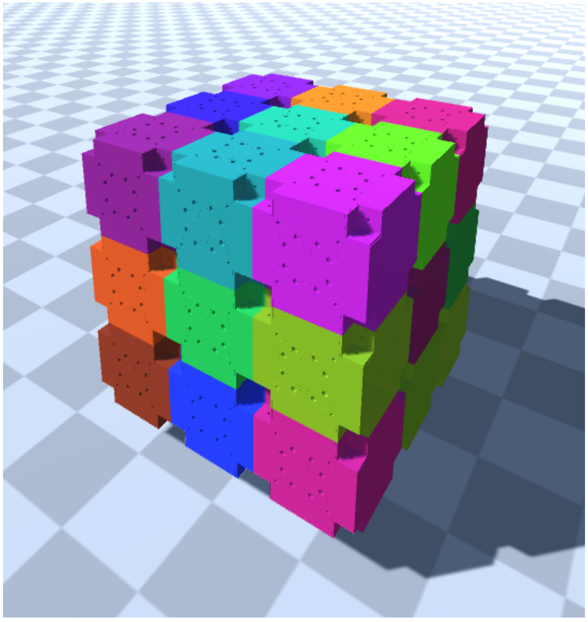


Figure 5.4: Render of a 3x3x3 strictly orthogonal network of modules in simulation.

5.5 Connector Mechanical Experiments

Mechanical experiments to assess the holding capacity of the Mo* connection mechanism were undertaken. These experiments give the holding force of the connector in the axis it will be loaded in during normal operation. These experiments determine characteristics such as how many modules can be held in a chain, for example.

Figure 3.8 in section 3.3 described the types of forces that are exerted between two modules. To evaluate the performance of the connection mechanism, an experiment was undertaken where one connection mechanism was rigidly held in place, and a force was exerted on a mating connector with a pulley while recording the exerted with a digital scale. The axial holding capacity was tested with the pull force being exerted on the center of the connector in the plane of the connecting face. The results of this experiment are depicted in figure 5.5. The maximum recorded load was 2.58 kg. The minimum recorded load was 2.21 kg. Over the 10 runs of the experiment, the mean load was 2.48 kg. This holding capacity would be sufficient to hold six modules vertically.

The next experiment undertaken was to test the bending load capacity of the connector. This is the holding capacity of one connection mechanism with respect to another when they are horizontally adjacent. This scenario would occur in kinematic chains of modules. One module was held in a fixed position. The connecting module was horizontally adjacent and the force was exerted at the center point of that module. This result can be regarded as a

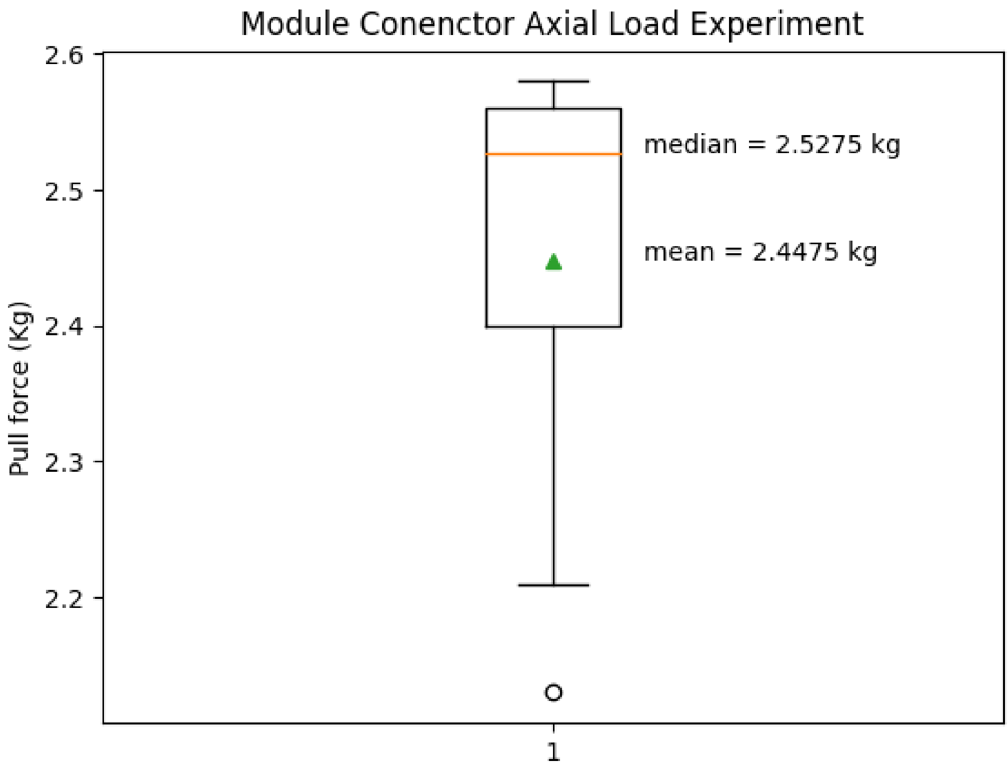


Figure 5.5: Results from axial load testing between two module connectors. The experiment was repeated n=10 times.

torque force, with the distance to the pivot point being 5 cm. The results of the experiment are shown in figure 5.6.

This experiment resulted in a maximum load of 0.935 kg and a minimum of 0.815 kg. The mean of the bending load was 0.992 kg. This is equivalent to a holding torque of 4.96Kg/cm. With this holding torque, a chain of two modules could be held horizontally from a connector.

The third experiment on module holding force undertaken was the twisting load. As with the bending load this can be given as a torque. Unlike the bending load, where the force is exerted at a normal to the face of the connection mechanism, the force in the twisting direction is exerted planar with the connection face. Scenarios where this would occur are in kinematic chains, where a chain of modules may be attached to the main body of the system in, for example, a legged configuration. Rather than acting on the pull force of the magnets as in the first two scenarios, this force acts against the sliding force of the magnets in the connection mechanism. The force was applied 5cm from the center of the connection

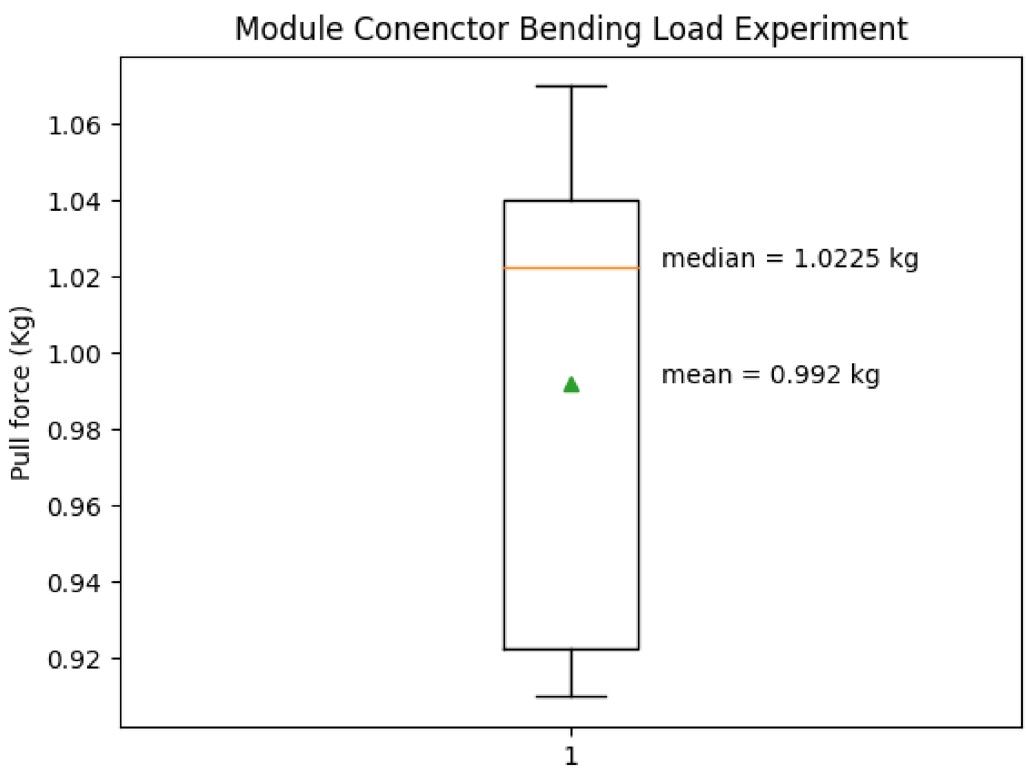


Figure 5.6: Results from bending load testing between two module connectors. The force was exerted at a distance of 5 cm from the connection face equivalent to the center of the module. The experiment was repeated n=10 times.

mechanism, equivalent to the edge of a module. The results of this experiment are shown in figure 5.7.

As with the others, the experiment was undertaken ten times to achieve an average result. The maximum load was 0.935 kg, and the minimum was 0.815 kg. The mean result of the experiment was 0.9 kg. With the load being applied at 5 cm from the pivot point, this results in a rotational holding torque of 4.5 kg/cm. The connection mechanism would hold a chain of two modules extended from the pivot module before failing.

These results quantify the holding capacity of the connection mechanism, which determines the structures that can be successfully created with the system without a connector mechanism failing. The results are summarised in table 5.1.

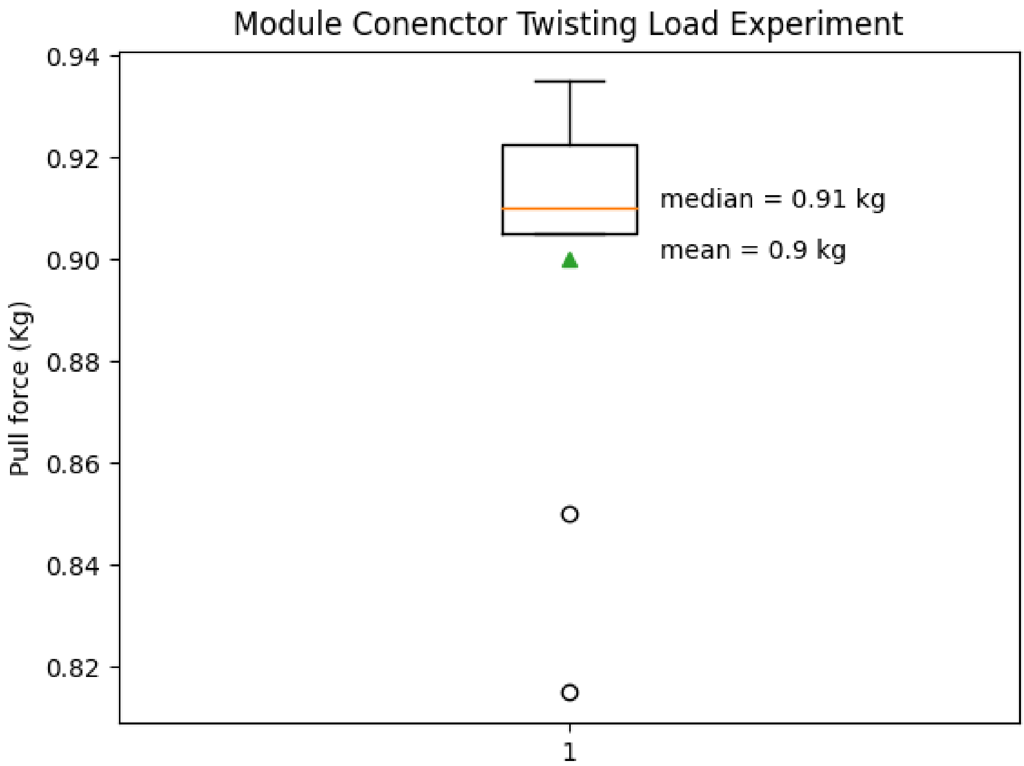


Figure 5.7: Results from twisting load testing between two module connectors. The force was exerted at a distance of 5 cm from the center connection face in the same plane. The experiment was repeated n=10 times.

Table 5.1: Summary of mechanical experiments on the Mo* connection mechanism.

Load	Mean Result
Axial	2.48 kg
Bending	4.96 kg/cm
Twisting	4.5 kg/cm

5.6 Electrical properties of the Mo* connection mechanism

Experiments were also performed on the electrical aspects of the connection mechanism. These include experiments on the absolute position sensing of the mechanism using the hall effect sensor embedded in each of the connector tile printed circuit boards, and the electrical performance of the Ethernet channel routed from the tile PCB through spring loaded contacts to the magnets.

5.6.1 Connector Absolute Position Sensing

Absolute position sensing is crucial for the successful operation of the rotating magnet connection mechanism. The quadrature encoder on the micro metal motor that drives the mechanism only provides relative positioning. For the homing sequence of the mechanism to operate successfully, absolute position must be known. In most rotating systems that require absolute positioning, an electro-mechanical limit switch is used. As space is a premium in the Mo* tile design, a small and preferably solid state implementation was required. Embedding a hall effect sensor on the bottom side of the tile PCB radially adjacent to the magnet home position provides this ability. Ideally the sensor would be placed axially in-line with the magnet, but this is not possible as the spring loaded contact is required to be in that position.

Figure 5.8 shows an oscilloscope view of the output of the hall effect sensor that is embedded on the slave PCB adjacent to the spring loaded contact. There is a 2.8mm gap between the sensing surface of the hall effect sensor and the closest side of the magnet in the pair when axially aligned. The result was a voltage swing of 1.09 V. This is more than sufficient for the ADC in the STM32f334 microcontroller on the slave board, or the ADC in the Zynq SoC to get a good dynamic range from.

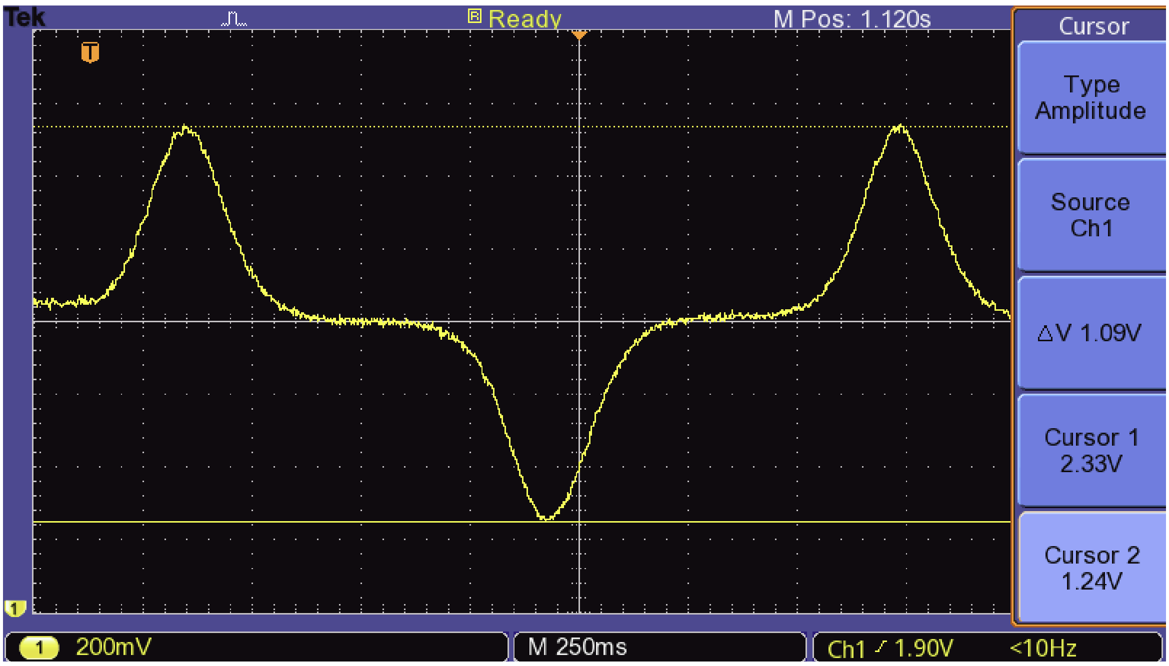


Figure 5.8: Oscilloscope view of the output of the hall effect sensor embedded on the tile PCB. The amplitude swing between the north and south aligned positions is measured.

Figure 5.9 depicts the measured time taken to transition between the active connect and disconnect states of the connector when driven by the micro metal motor. This is equivalent to a 180 degree rotation of the magnet pair. The time taken to transition is 0.81 seconds.

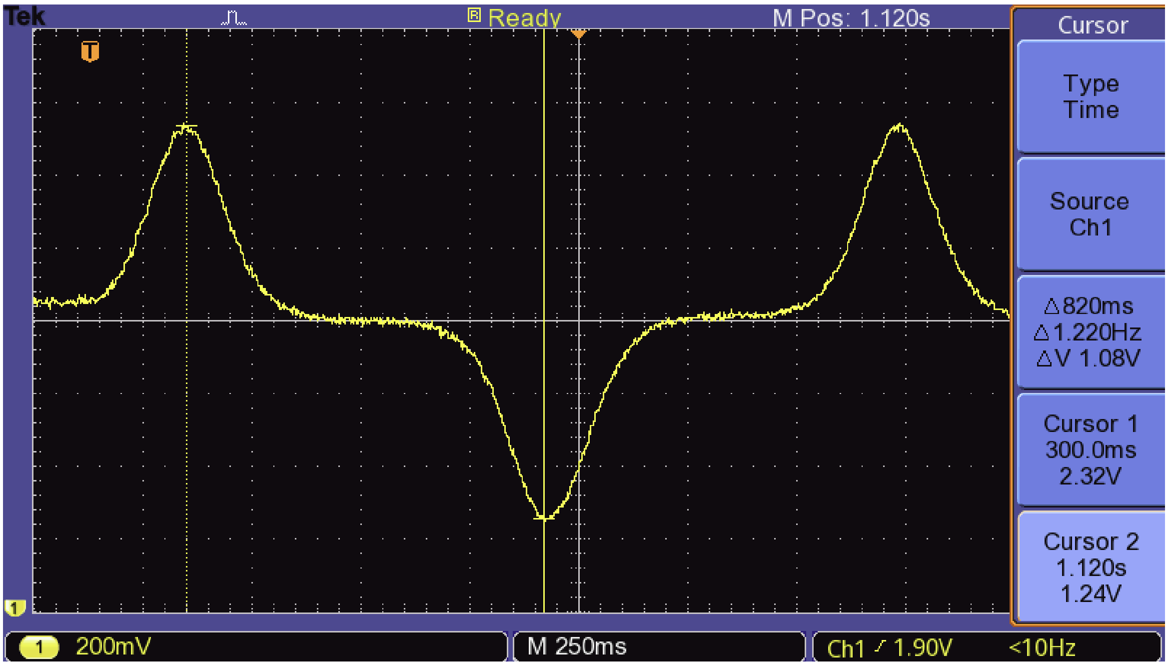


Figure 5.9: Oscilloscope view of the output of the hall effect sensor embedded on the tile PCB. The time between north and south aligned positions is measured.

Figure 5.10 depicts the output of the hall effect sensor with the north pole and south pole aligned magnet positions labelled. The 180 degree transition period in the measurement is shaded in green. With the turning points of this output from the hall effect sensor, the module connector's absolute position can be determined at the active connect and active disconnect states. The quadrature encoder on the micro metal motor is used to keep track of where within the region the connector is between connection states.

5.6.2 Ethernet Channel Electrical Characteristics

The line level electrical characteristics of the Ethernet channel between the transformers of two modules are an important factor for successful operation of the 100Base-T Ethernet Communication. Normally an Ethernet channel would consist of the PCB trace from the transformer to an RJ45 connector, a twisted-pair Ethernet cable, and then the same on the other end of the connection in reverse. In the Mo* module system, from one transformer to another, the Ethernet channel is composed of the PCB trace between the transformer and the spring loaded contact, the spring loaded contact itself, the magnet on each of the connecting

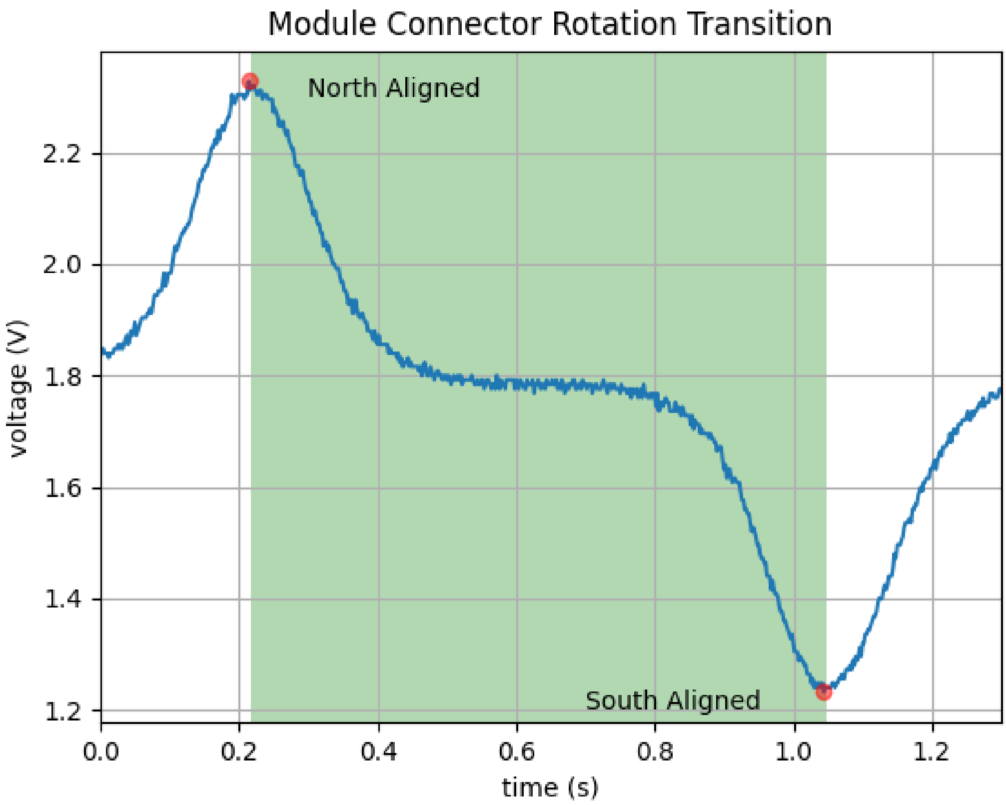


Figure 5.10: Depiction the output of the hall effect sensor through a connector rotation of 180 degrees, or active connect to active disconnect. The north aligned and south aligned points are highlighted. Shaded area depicts the 180 degree transition region.

faces, and the spring loaded contact and PCB trace to the transformer on the connecting module. To evaluate the electrical channel, the DC impedance was taken between the line side of the transformers on two connected modules. The results of these experiments are shown in figure 5.11. The impedance of each of the Ethernet signal lines (TD_P,TD_N,RD_P,RD_N) are 151.6Ω,151.2Ω,150.8Ω,150.6 Ω. The transmit differential pair has a difference of 0.4 Ω between TD_P and TD_N. The receive differential pair has a difference of 0.2 Ω between RD_P and RD_N.

5.7 Chapter Summary

This chapter has detailed the implementation of the simulation environment created to model various aspects of the Mo* modular robot platform and the experiments undertaken to evaluate the mechanical and electrical performance of the Mo* module connection mechanism. The

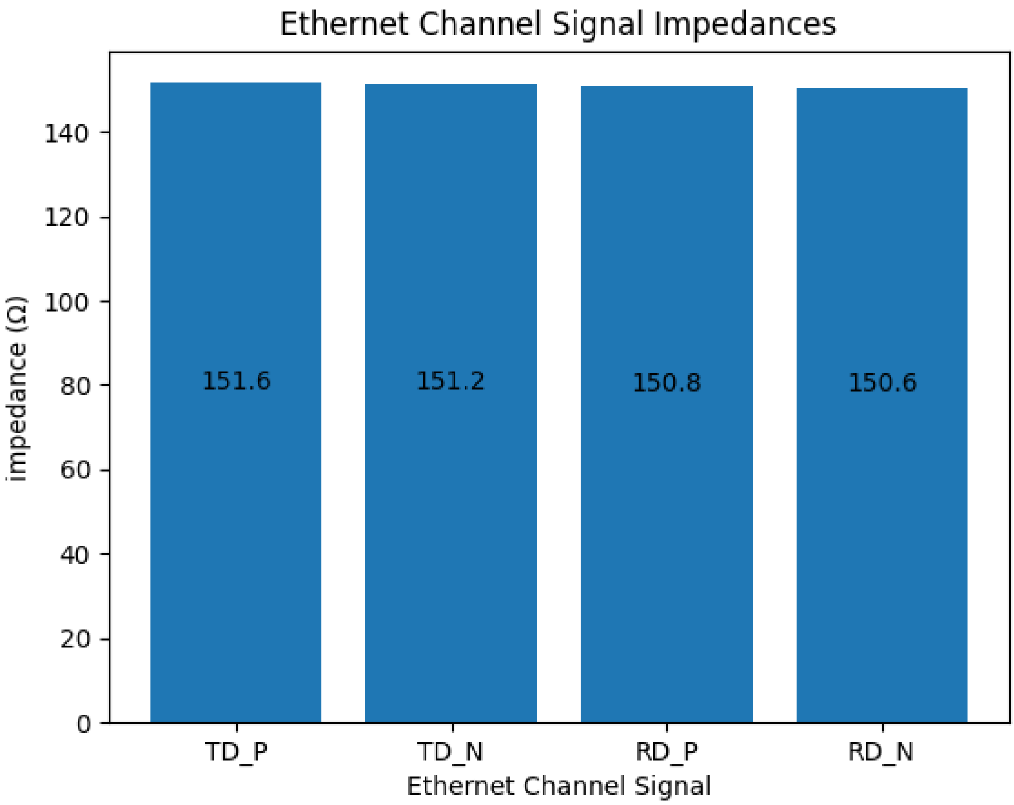


Figure 5.11: DC impedance of each of the Ethernet channel data signal lines when measured between the line side of transformers of adjacent modules.

simulation models the physical environment that the modules operate in, the network channel between modules, the operation of various aspects of the module master PCB including the PHY, FPGA functions, and program control code. The failure load of the mechanism was tested. The connection mechanism is on average resistant to loads of 2.48 kg axially, 4.96 kg/cm bending at normal to connection face, and 4.5 kg/cm twisting planar to the connection face. The connection mechanism is strongest in the axial direction. The bending and twisting load directions are close to equally strong. With the holding strength of this connector, complex morphologies of modules could be constructed. Experiments have been undertaken to assess the performance of the absolute position sensing capabilities of the connection mechanism. This showed that the hall effect sensor can effectively and accurately obtain the absolute position of the connection mechanism when one of the magnets in the pair are aligned with it. This is a neat and low component count solution to the problem of absolute position sensing. Experiments have also been undertaken on the electrical properties of the connection mechanism. The DC line impedance of each of the data lines in the Ethernet channel between two modules was measured. All of the data lines measured a non-infinite impedance, meaning the electrical connection between modules is working successfully. All of the pairs are within 0.4 Ω of each other, which demonstrates the effective differential pair length matching within a module.

Chapter 6

Conclusions

Research into modular robotics has evolved substantially in recent years. Work has been undertaken by various research institutions around the globe to implement modular robotic platforms for various applications. Much of this work has been focused on developing particular aspects of modular robot systems such as reconfiguration algorithms, connection mechanisms, and locomotion strategies. While all of this work has been valuable, it has resulted in a plethora of modular robot platforms that are task specific, rather than generally applicable. To get modular robots out of the research lab and in to real world scenarios, work is needed to implement a generally capable modular robot platform that can be used as a standardised framework for creating the next generation of modular robot systems. As has been proven with implemented standards in the communications, IT hardware and manufacturing sectors in the past, implementing general standards greatly speeds up the time from concept to implementation of new innovations. This work aims to go some way to creating a new standard for modular robot systems that can be used to impact the speed at which new research is implemented.

The work in this thesis has described the design, implementation and testing of a new modular robot platform called Mo* that can be generally applicable and easily extensible to a wide variety of research and real world applications. The platform has a 10x10x10cm cubic structure. The modules contain up to six active connection mechanisms, allowing for three dimensional lattice structures to be created.

The platform includes a novel module connection mechanism based on rotating pairs of permanent magnets. The connection mechanism is genderless, can perform single sided disconnection, contains no protruding components, and only requires power draw when transitioning between connected and unconnected states.

The platform also includes a communication strategy between modules that uses the magnets in the module connection mechanism as the electrical conductors. It uses the IEE802.3 100Base-T Ethernet standard as the physical layer for communication, giving speeds of up to 100Mb/s between modules. The communication strategy also includes a packet switched communication protocol for routing data between modules.

A simulation environment has also been implemented that provides a complete system for simulating systems of modular robots. This simulator includes physical, networking and control simulation of the system and its architecture is designed to be extensible to heterogeneous modules with varying sensing and actuation capabilities.

6.1 Discussion

This discussion addresses the achievements of the system, the limitations identified during the course of the project, and finally some points of future development and research are identified and discussed. The current version of the Mo* modular system can be assessed against the requirements set out at the beginning of this thesis. On the whole, the author believes that this thesis lays a groundwork for developing a truly general heterogeneous modular robot platform, which can be used for a wide variety and research applications. The system is simple to produce with off the shelf electronic components and 3D printing facilities.

6.1.1 Achievements of the Mo* modular system

The Mo* modular system was implemented to demonstrate the capabilities and uses of a generally applicable modular robot platform. This platform aims to be an open standard that can be used to accelerate the development of modular robot capabilities by providing a base platform that can be used for a wide variety of experiments and applications. There are a number of specific aspects of the system that are worth particular mention.

- **Magnetic connection mechanism** - A module connection mechanism based on rotating pairs of magnets has been implemented. This uses eight 5mm neonydium maagnets arranged in to four rotating pairs. This effectively creates a semi-permanent magnetic connection between the modules where the total holding force when in the connection state is equal to the sum of the holding forces of each of the magnets. No power is required when in a holding state, only when the mechanism is transitioning from active connect to active disconnect states.

- **Sub modular master-slave tile structure** - The system was designed around a master-slave architecture. Each module contains one master tile which is responsible for all of the major control and networking functionality of the system. There can be up to five slave tiles in each module. These are connected to the master tile through a tightly defined interface which enables heterogeneous modules to be easily constructed by implementing a heterogeneous tile and integrating that in to existing module hardware. Currently each slave tile contains an STM32F334 MCU which controls the connection mechanism for its tile and offers GPIO expansion for function extendibility.

- **Module extendibility** - The master slave architecture implements a new paradigm for design and construction of heterogeneous modular systems. It allows heterogeneous modules to be easily implemented. This will greatly reduce the time taken for researchers to implement their own heterogeneous modules in the future, as only one tile has to be designed and built, then integrated with existing tile designs.

- **100Mb/s packet switched communication between modules** - A communication strategy based on 100Base-T Ethernet physical layer was implemented. This uses the magnets in the connector to provide the electrical connection and route the two differential data pairs for the Ethernet channel between two modules in any orientation. Each module can have up to six communication-enabled tiles, which connenct to the Zynq SoC on the master tile which acts as a programmable network switch through the FPGA fabric on the device.

- **Modular system simulation platform** - A simulation platform for dynamically reconfigurable modular systems has been implemented. This simulator is capable of modelling the physical, networking and control aspects of the modular system in definable environments. This platform allows researchers to conduct experiments on varying aspects modular robotic systems without the need to first create hardware prototypes.

6.1.2 Limitations of the Mo* modular system

- **Power sharing implementation** - While there is provision in the electronics for sharing power over the Ethernet channel similar to that of the IEEE802.3af specification, a full implementation and testing of this system was never performed. At current each module needs to source it's own power, either from an on board battery or from an external power supply. It would greatly enhance the system's capabilities if this implementation was completed and tested.

- **Heterogeneous modules implementation** - Some work was done to implement a wheel module in the system, but this was not completed and tested within the time frame of the project. Giving the system locomotion capabilities would aid in demonstrating the capabilities of modular systems in real world settings. Multiple rover configurations could be implemented to give the system the ability to adapt to different terrains.

- **Experiments on Mo* network protocol** - Due to time constraints, it was not possible to obtain data from hardware to test the network protocol during the course of this work. This is quite a large gap in validating the Mo* protocol completely. This work should be undertaken before developing the system further.

- **Programming difficulty** - Currently each of the tiles in a module need to have program code loaded on to them individually. This involves five programming operations for the slave tiles, with the code being the same apart from the CAN bus ID which is individually assigned for each tile. The master tile needs to be programmed seperatly from Xilinx Vivado tools. Work was done to implement programming of the slave tiles from the master tile, and loading new program code on to the master tile through the external ethernet interface. But this implementation was not completed.

- **PLA material construction** - The PLA material used for all of the 3D printed parts has worked well for testing the system, and while all of the printed parts proved mechanically strong enough during testing, there are some concerns about the durability of PLA over time. Using a material such as ABS or PET-G could help give more durability to the modules.

6.1.3 Future Work

There are many directions further development of the Mo* system could be taken, as well as the field of modular robotics as a whole. Some of the notable points are outlined here.

Currently, no sensor tiles have been created for the Mo* system. There are a wide variety of sensor tiles that could be created for varying applications. Autonomous navigation of a modular rover could be achieved with sensing capabilities such as camera, LIDAR or IMU tiles. Tiles to conduct scientific measurements gathering environmental data could also be created. One can imagine this in the context of a planetary exploration modular rover. Some such tiles that could be developed include temperature,humidity,pressure, atmospheric light, and radiation sensors.

An array of kinematic tiles and modules could be developed for the Mo* system. Wheel tiles could enable reconfigurable rovers to be created. Modules that allow even one rotational degree of freedom between two connection mechanisms could be connected together in a chain configuration to create complex mechanical structures with many degrees of freedom. Linear actuation modules could also be developed. This could enable configurations where active suspension could be applied to modular rovers, or sensor turrets that can accurately set their position relative to the target or help maintain sensor stability during locomotion.

The electrical components for power sharing over the Ethernet channel similar to that of the IEEE802.3af standard. This allows a DC voltage differential to be applied between the data pairs. A system could be developed where certain modules are net producers of energy, containing large battery reserves. They could then supply power to other modules in the system that are net power consumers. Intelligent power routing schemes could be developed to efficiently share power around the system. This could enable modules that have excess energy, or have a lower mission priority, to share power to modules that are critical to mission success. This could create an intelligent power grid system with similar properties to national electrical grids, ensuring power is supplied to the correct parts of the system to ensure mission success.

The applications of modular robots are wide ranging. One particular area of interest is getting modular robot systems to space. A system could be implemented where there is a storage facility of heterogeneous modules in orbit. If a modular robotic system was exploring the moon or mars for example, and a module on the system ceased to function, a replacement could be sent from the orbiting store of modules. The system could then reconfigure itself by ejecting the broken module using the single sided disconnect mechanism, and attach the replacement module, regaining the ability to continue its mission. This would become particularly efficient if there was a large number of modular systems undertaking various tasks in the near solar system, as the cost and time to transport a module from orbit to the system in question would be a lot less and if it had to be sent from earth.

Modular robots could be very useful in educational scenarios. A system where students can easily assemble components of a robotic system (modules in this case) and quickly get a hardware platform constructed would reduce the time to implementing software on the system. Students could exercise creativity in designing system configurations with heterogeneous modules to accomplish a variety of tasks.

Bibliography

[1] “Cabridge dictionary definition of "robot".” https://dictionary.cambridge.org/dictionary/english/robot. Accessed: 2023-09-19.

[2] T. Fukuda and Y. Kawauchi, “Cellular robotic system (CEBOT) as one of the realization of self-organizing intelligent universal manipulator,” pp. 662–667, 1990.

[3] M. A. Post, X.-T. Yan, and P. Letier, “Modularity for the future in space robotics: A review,” *Acta Astronaut.*, vol. 189, pp. 530–547, Dec. 2021.

[4] M. Yim, B. Shirmohammadi, J. Sastra, M. Park, M. Dugan, and C. J. Taylor, “Towards robotic self-reassembly after explosion,” *Rep. U. S.*, pp. 2767–2772, 2007.

[5] M. Yim, W.-M. Shen, B. Salemi, D. Rus, M. Moll, H. Lipson, E. Klavins, and G. S. Chirikjian, “Modular Self-Reconfigurable robot systems [grand challenges of robotics],” *IEEE Robot. Autom. Mag.*, vol. 14, pp. 43–52, Mar. 2007.

[6] M. Yim, K. Roufas, D. Duff, Y. Zhang, C. Eldershaw, and S. Homans, “Modular reconfigurable robots in space applications,” *Auton. Robots*, 2003.

[7] P. Moubarak and P. Ben-Tzvi, “Modular and reconfigurable mobile robotics,” Dec. 2012.

[8] A. Brunete, A. Ranganath, S. Segovia, J. P. de Frutos, M. Hernando, and E. Gambao, “Current trends in reconfigurable modular robots design,” *Int. J. Adv. Rob. Syst.*, vol. 14, p. 172988141771045, May 2017.

[9] H. Ahmadzadeh, E. Masehian, and M. Asadpour, “Modular robotic systems: Characteristics and applications,” *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 81, no. 3-4, pp. 317–357, 2016.

[10] J. Feczko, M. Manka, P. Krol, M. Giergiel, T. Uhl, and A. Pietrzyk, “Review of the modular self reconfigurable robotic systems,” *2015 10th International Workshop on Robot Motion and Control, RoMoCo 2015*, pp. 182–187, 2015.

[11] M. Yim, C. Eldershaw, Y. Zhang, and D. Duff, “Self-Reconfigurable robot systems: PolyBot,” *Adv. Robot.*, vol. 21, no. 8, pp. 851–854, 2003.

[12] C. Parrott, “A hybrid and extendable self-reconfigurable modular robotic system,” *PQDT - UK & Ireland*, no. September, 2016.

[13] R. Groß and M. Dorigo, "Self-assembly at the macroscopic scale," *Proc. IEEE*, vol. 96, no. 9, pp. 1490–1508, 2008.

[14] J. Feczko, M. Manka, P. Król, and A. Pietrzyk, "Review of the modular self reconifgurabe robotic systems," in *RoMoCo 2015*, unknown, July 2015.

[15] R. Alattas, "Evolutionary modular Self-Assembly & Self-Reconfigurable robotics: Exhaustive review," *arXiv e-prints*, 2017.

[16] E. Del Dottore, A. Sadeghi, A. Mondini, V. Mattoli, and B. Mazzolai, "Toward growing robots: A historical evolution from cellular to plant-inspired robotics," *Frontiers Robotics AI*, vol. 5, no. MAR, 2018.

[17] W. Saab, P. Racioppo, and P. Ben-Tzvi, "A review of coupling mechanism designs for modular reconfigurable robots," *Robotica*, vol. 37, no. 2, pp. 378–403, 2019.

[18] M. Yim, "New locomotion gaits," *Proceedings - IEEE International Conference on Robotics and Automation*, no. pt 3, pp. 2508–2514, 1994.

[19] A. Castano, W. M. Shen, and P. Will, "CONRO: towards deployable robots with inter-robot metamorphic capabilities," *Auton. Robots*, vol. 8, no. 3, pp. 309–324, 2000.

[20] D. Marbach and A. J. Ijspeert, "Online optimization of modular robot locomotion," *IEEE International Conference on Mechatronics and Automation, ICMA 2005*, vol. 1, no. July, pp. 248–253, 2005.

[21] B. Salemi, M. Moll, and W. M. Shen, "SUPERBOT: A deployable, multi-functional, and modular self-reconfigurable robotic system," in *IEEE International Conference on Intelligent Robots and Systems*, pp. 3636–3641, 2006.

[22] J. Liedke, R. Matthias, L. Winkler, and H. Worn, "The collective self-reconfigurable modular organism (CoSMO)," *2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics: Mechatronics for Human Wellbeing, AIM 2013*, pp. 1–6, 2013.

[23] L. Pfotzer, S. Ruehl, G. Heppner, A. Roennau, and R. Dillmann, "KAIRO 3: A modular reconfigurable robot for search and rescue field missions," *2014 IEEE International Conference on Robotics and Biomimetics, IEEE ROBIO 2014*, pp. 205–210, 2014.

[24] S. Murata, H. Kurokawa, and S. Kokaji, "Self- as sembling machine," pp. 3–10, 1994.

[25] J. W. Romanishin, K. Gilpin, and D. Rus, “M-blocks: Momentum-driven, magnetic modular robots,” *Rep. U. S.*, pp. 4288–4295, 2013.

[26] T. Fukuda and S. Nakagawa, “Dynamically reconfigurable robotic system,” in *Proceedings. 1988 IEEE International Conference on Robotics and Automation*, pp. 1581–1586 vol.3, Apr. 1988.

[27] T. Fukuda, T. Ueyama, and F. Arai, “Communication strategy for network of cellular robotic system (CEBOT)-mastering methods for single and two tops structures,” pp. 527–532 vol.1, 1991.

[28] D. Rus and M. Vona, “Physical implementation of the self-reconfiguring crystalline robot,” *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2, no. April, pp. 1726–1733, 2000.

[29] Daniela Rus and Marsette Vona, “Crystalline robots: Self-reconfiguration with compressible unit modules,” *Auton. Robots*, vol. 10, no. 1, pp. 107–124, 2001.

[30] A. Castano, A. Behar, and P. M. Will, “The conro modules for reconfigurable robots,” *IEEE/ASME Trans. Mechatron.*, vol. 7, no. 4, pp. 403–409, 2002.

[31] W. M. Shen, B. Salemi, and P. Will, “Hormone-inspired adaptive communication and distributed control for CONRO self-reconfigurable robots,” *IEEE Trans. Rob. Autom.*, 2002.

[32] G. S. Chirikjian, “Metamorphic Hyper-Redundant manipulators,” 1993.

[33] I. M. Chen and J. W. Burdick, “Determining task optimal modular robot assembly configurations,” *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 1, no. October, pp. 132–137, 1995.

[34] G. J. Hamlin and A. C. Sanderson, “TETROBOT modular robotics: prototype and experiments,” *Rep. U. S.*, vol. 2, pp. 390–395, 1996.

[35] A. P. Chirikjian, D. Stein, C.-J. Chiang, and Grigory, “Design and implementation of metamorphic robots,” in *Proceedings of The ASME Design Engineering Technical Conference and Computers in Engineering Conference*, pp. 1–10, 1996.

[36] K. Kotay, D. Rus, M. Vona, and C. McGray, “The self-reconfiguring robotic molecule,” *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 1, pp. 424–431, 1998.

[37] S. Murata, H. Kurokawa, E. Yoshida, K. Tomita, and S. Kokaji, "A 3-D Self-Reconfigurable structure," no. May, 1998.

[38] C. Unsal and P. K. Khosla, "Solutions for 3D self-reconfiguration in a modular robotic system: implementation and motion planning," *Sensor Fusion and Decentralized Control in Robotic Systems III*, vol. 4196, no. November, pp. 388–401, 2000.

[39] M. Yim, D. G. Duff, and K. D. Roufas, "PolyBot: a modular reconfigurable robot," in *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 1, pp. 514–520, 2000.

[40] S. Murata, E. Yoshida, A. Kamimura, H. Kurokawa, K. Tomita, and S. Kokaji, "M-TRAN : Self-Reconfigurable modular," vol. 7, no. 4, pp. 431–441, 2002.

[41] J. W. Suh, S. B. Homans, and M. Yim, "Telecubes: Mechanical design of a module for self-reconfigurable robotics," *Proceedings-IEEE International Conference on Robotics and Automation*, vol. 4, no. May, pp. 4095–4101, 2002.

[42] H. Karbasi, J. P. Huissoon, and A. Khajepour, "Uni-drive modular robots: Theory, design, and experiments," *Mechanism and Machine Theory*, vol. 39, no. 2, pp. 183–200, 2004.

[43] M. W. Jørgensen, E. H. Østergaard, and H. H. Lund, "Modular ATRON: Modules for a self-reconfigurable robot," *Rep. U. S.*, vol. 2, pp. 2068–2073, 2004.

[44] Y. Terada and S. Murata, "Automatic assembly system for a large-scale modular structure - hardware design of module and assembler robot," *Rep. U. S.*, vol. 3, pp. 2349–2355, 2004.

[45] M. Koseki, K. Minami, and N. Inou, "Cellular robots forming a mechanical structure (evaluation of structural formation and hardware design of " CHOBIE II ")," *Proceedings of 7th International Symposium on Distributed Autonomous Robotic Systems (DARS04)*, pp. 131–140, 2004.

[46] P. J. White, K. Kopanski, and H. Lipson, "Stochastic self-reconfigurable cellular robotics," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2004, no. 3, pp. 2888–2893, 2004.

[47] D. Marbach, "Evolution and online optimization of central pattern generators for modular robot locomotion," *Unpublished Master Thesis, Swiss Federal Institute of . . .*, 2004.

[48] F. Mondada, G. C. Pettinaro, A. Guignard, I. W. Kwee, D. Floreano, J. L. Deneubourg, S. Nolfi, L. M. Gambardella, and M. Dorigo, “Swarm-bot: A new distributed robotic concept,” *Auton. Robots*, vol. 17, no. 2-3, pp. 193–221, 2004.

[49] A. Brunete, M. Hernando, and EGambao, “Modular multiconfigurable architecture for low diameter pipe inspection microrobot,” *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2005, no. April, pp. 490–495, 2005.

[50] E. Schweikardt and M. D. Gross, “RoBlocks: A robotic construction kit for mathematics and science education,” *ICMI’06: 8th International Conference on Multimodal Interfaces, Conference Proceeding*, pp. 72–75, 2006.

[51] M. Shimizu, T. Mori, and A. Ishiguro, “A development of a modular robot that enables adaptive reconfiguration,” *Rep. U. S.*, pp. 174–179, 2006.

[52] V. Zykov and H. Lipson, “Experiment design for stochastic Three-Dimensional reconfiguration of modular robots,” *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pp. 2–5, 2007.

[53] D. Daidié, O. Barbey, A. Guignard, D. Roussy, F. Guenter, A. Ijspeert, and A. Billard, “The DoF-box project: An educational kit for configurable robots,” *IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM*, 2007.

[54] K. Gilpin, K. Kotay, D. Rus, and I. Vasilescu, “Miche: Modular shape formation by Self-Disassembly,” *Int. J. Rob. Res.*, vol. 27, pp. 345–372, Mar. 2008.

[55] A. Sproewitz, A. Billard, P. Dillenbourg, and A. J. Ijspeert, “Roombots-mechanical design of self-reconfiguring modular robots for adaptive furniture,” pp. 4259–4264, 2009.

[56] H. Zhang, J. Gonzalez-Gomez, Z. Xie, S. Cheng, and J. Zhang, “Development of a low-cost flexible modular robot GZ-I,” *IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM*, pp. 223–228, 2008.

[57] A. Lyder, R. F. M. Garcia, and K. Stoy, “Mechanical design of odin, an extendable heterogeneous deformable modular robot,” *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, pp. 883–888, 2008.

[58] A. Yerpes, J. Baca, J. A. Escalera, M. Ferre, R. Aracil, and A. Crespo, “Modular robot based on 3 rotational DoF modules,” *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, pp. 889–894, 2008.

[59] H. Wei, Y. Cai, H. Li, D. Li, and T. Wang, "Sambot: A self-assembly modular robot for swarm robot," *Proceedings - IEEE International Conference on Robotics and Automation*, no. 37, pp. 66–71, 2010.

[60] G. G. Ryland and H. H. Cheng, "Design of imobot, an intelligent reconfigurable mobile robot with novel locomotion," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 60–65, 2010.

[61] K. Gilpin, A. Knaian, and D. Rus, "Robot pebbles: One centimeter modules for programmable matter through self-disassembly," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 2485–2492, 2010.

[62] A. Lyder, R. Franco, M. Garcia, and K. Stoy, "Genderless connection mechanism for modular robots introducing torque transmission between modules," *ICRA 2010 Workshop on Modular Robots: The State of the Art*, pp. 77–82, 2010.

[63] Y. Meng, Y. Zhang, A. Sampath, Y. Jin, and B. Sendhoff, "Cross-ball: A new morphogenetic self-reconfigurable modular robot," *Proceedings - IEEE International Conference on Robotics and Automation*, no. 1, pp. 267–272, 2011.

[64] R. Oung and R. D'Andrea, "The distributed flight array," *Mechatronics*, vol. 21, no. 6, pp. 908–917, 2011.

[65] J. Zhao, X. Cui, Y. Zhu, and S. Tang, "A new self-reconfigurable modular robotic system UBot: Multi-mode locomotion and self-reconfiguration," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 150080, no. 2, pp. 1020–1025, 2011.

[66] K. C. Wolfe, M. S. Moses, M. D. M. Kutzer, and G. S. Chirikjian, "M3Express: A low-cost independently-mobile reconfigurable modular robot," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 2704–2710, 2012.

[67] J. Baca, M. Ferre, and R. Aracil, "A heterogeneous modular robotic design for fast response to a diversity of tasks," *Rob. Auton. Syst.*, vol. 60, no. 4, pp. 522–531, 2012.

[68] J. Davey, N. Kwok, and M. Yim, "Emulating self-reconfigurable robots - design of the SMORES system," *Rep. U. S.*, pp. 4464–4469, 2012.

[69] B. Piranda, G. J. Laurent, C. Clévy, S. Möbes, and N. L. Fort-Piat, "A new concept of planar self-reconfigurable modular robot for conveying microparts," *Mechatronics*, vol. 23, pp. 906–915, Oct. 2013.

[70] M. Dorigo, D. Floreano, L. M. Gambardella, F. Mondada, S. Nolfi, T. Baaboura, M. Birattari, M. Bonani, M. Brambilla, A. Brutschy, D. Burnier, A. Campo, A. L. Christensen, A. Decugniere, G. Di Caro, F. Ducatelle, E. Ferrante, A. Förster, J. M. Gonzales, J. Guzzi, V. Longchamp, S. Magnenat, N. Mathews, M. Montes De Oca, R. O'Grady, C. Pinciroli, G. Pini, P. Rétornaz, J. Roberts, V. Sperati, T. Stirling, A. Stranieri, T. Stützle, V. Trianni, E. Tuci, A. E. Turgut, and F. Vaussard, "Swarmanoid: A novel concept for the study of heterogeneous robotic swarms," *IEEE Robot. Autom. Mag.*, vol. 20, no. 4, pp. 60–71, 2013.

[71] Y. Zhang, G. Song, S. Liu, G. Qiao, J. Z. J. o. I. . . . ., and U. 2016, "A modular self-reconfigurable robot with enhanced locomotion performances: design, modeling, simulations, and experiments," *Springer*, 2014.

[72] R. Thakker, A. Kamat, S. Bharambe, S. Chiddarwar, and K. M. Bhurchandi, "ReBiS - reconfigurable bipedal snake robot," *Rep. U. S.*, no. Iros, pp. 309–314, 2014.

[73] B. Salem, "PetRo: Development of a modular pet robot," *Proceedings - IEEE International Workshop on Robot and Human Interactive Communication*, vol. 2014-Octob, no. October, pp. 483–488, 2014.

[74] J. Baca, S. G. M. Hossain, P. Dasgupta, C. A. Nelson, and A. Dutta, "ModRED: Hardware design and reconfiguration planning for a high dexterity modular self-reconfigurable robot for extra-terrestrial exploration," *Rob. Auton. Syst.*, vol. 62, no. 7, pp. 1002–1015, 2014.

[75] K. J. Gucwa and H. H. Cheng, "RoboSim: A simulated environment for programming modular robots," *MESA 2014 - 10th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications, Conference Proceedings*, 2014.

[76] L. Pfotzer, J. Oberlaender, A. Roennau, and R. Dillmann, "Development and calibration of KaRoLa, a compact, high-resolution 3D laser scanner," in *2014 IEEE International Symposium on Safety, Security, and Rescue Robotics (2014)*, pp. 1–6, Oct. 2014.

[77] V. Kee, N. Rojas, M. R. Elara, and R. Sosa, "Hinged-Tetro : A self-recon fi gurable module for nested recon fi guration," pp. 1539–1546, 2014.

[78] I. O'Hara, J. Paulos, J. Davey, N. Eckenstein, N. Doshi, T. Tosun, J. Greco, J. Seo, M. Turpin, V. Kumar, and M. Yim, "Self-assembly of a swarm of autonomous boats

into floating structures," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 1234–1240, 2014.

[79] X. Jia, M. Frenger, Z. Chen, W. R. Hamel, and M. Zhang, "An alligator inspired modular robot," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2015-June, no. June, pp. 1949–1954, 2015.

[80] A. Faiña, F. Bellas, F. Orjales, D. Souto, and R. J. Duro, "An evolution friendly modular architecture to produce feasible robots," *Rob. Auton. Syst.*, vol. 63, no. P2, pp. 195–205, 2015.

[81] J. Neubert and H. Lipson, "Soldercubes: a self-soldering self-reconfiguring modular robot system," *Auton. Robots*, vol. 40, no. 1, pp. 139–158, 2016.

[82] M. Pacheco, R. Fogh, H. H. Lund, and D. J. Christensen, "Fable II: Design of a modular robot for creative learning," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2015-June, no. June, pp. 6134–6139, 2015.

[83] M. J. Doyle, X. Xu, Y. Gu, F. P.-D. . R. . . . , , and U. 2016, "Modular hydraulic propulsion: A robot that moves by routing fluid through itself," *ieeexplore.ieee.org*.

[84] "Raspberry pi foundation." https://www.raspberrypi.com/. Accessed: 2023-08-23.

[85] "Arduino." https://www.arduino.cc/. Accessed: 2023-08-23.

[86] "Raspberry pi 4 model b." https://www.raspberrypi.com/products/raspberry-pi-4-model-b/. Accessed: 2023-08-23.

[87] "Arduino classic family." https://www.arduino.cc/en/hardware#classic-family. Accessed: 2023-08-23.

[88] "Mag-tools europe." mag-tools.eu. Accessed: 2023-08-23.

[89] "Tower pro." https://www.towerpro.com.tw/product/sg90-7/. Accessed: 2023-08-27.

[90] "Pololu robotics and electronics, micro metal motors." https://www.pololu.com/category/60/micro-metal-gearmotors. Accessed: 2023-08-27.

[91] "First 4 magnets uk." https://www.first4magnets.com. Accessed: 2023-08-27.

[92] "Roymech tribology." https://www.roymech.co.uk/Useful_Tables/Tribology/co_of_frict.htm#coef. Accessed: 2023-08-27.

[93] "Harwin pogo pins." https://www.harwin.com/connectors-hardware/spring-contacts/pogo-pins/. Accessed: 2023-08-27.

[94] "U.s. patent 6,175,865." https://patents.google.com/patent/WO2001077699A1/en. Accessed: 2023-08-29.

[95] "Trenz te0720, trenz electronic gmbh." https://shop.trenz-electronic.de/en/Products/Trenz-Electronic/TE07XX-Zynq-SoC/TE0720-Zynq-SoC/. Accessed: 2023-09-12.

[96] "Samtec razor beam fine pitch self mating conenctors." https://www.samtec.com/connectors/high-speed-board-to-board/mezzanine/razor-beam. Accessed: 2023-09-12.

[97] "Rmii interface specification, rmii consortium." http://ebook.pldworld.com/_eBook/-Telecommunications,Networks-/TCPIP/RMII/rmii_rev12.pdf. Accessed: 2023-09-19.

[98] M. Santos Pessoa de Melo, J. Gomes da Silva Neto, P. Jorge Lima da Silva, J. M. X. Natario Teixeira, and V. Teichrieb, "Analysis and comparison of robotics 3d simulators," in *2019 21st Symposium on Virtual and Augmented Reality (SVR)*, pp. 242–251, 2019.

[99] "Simulation in robotics," *Math. Comput. Simul.*, vol. 79, pp. 879–897, Dec. 2008.

[100] A. Staranowicz and G. L. Mariottini, "A survey and comparison of commercial and open-source robotic simulator software," in *Proceedings of the 4th International Conference on PErvasive Technologies Related to Assistive Environments*, pp. 1–8, 2011.

[101] L. Žlajpah, "Simulation in robotics," *Mathematics and Computers in Simulation*, vol. 79, no. 4, pp. 879–897, 2008.

[102] "Gazebo simulation platform." https://gazebosim.org/home. Accessed: 2023-09-19.

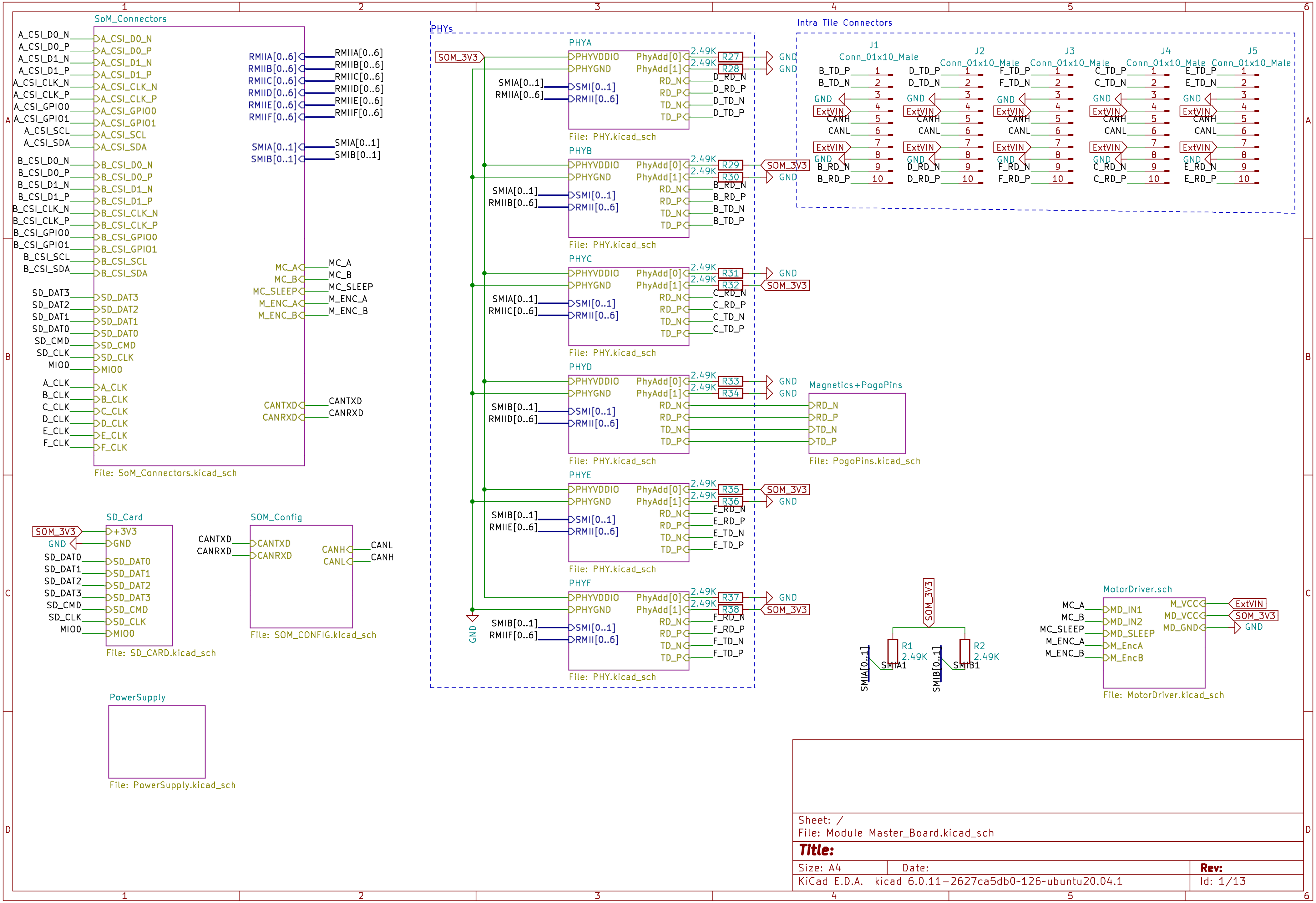
[103] "ns3 network simulator." https://www.nsnam.org/. Accessed: 2023-09-19.

[104] "Qemu emulation platform." https://www.qemu.org/. Accessed: 2023-09-19.

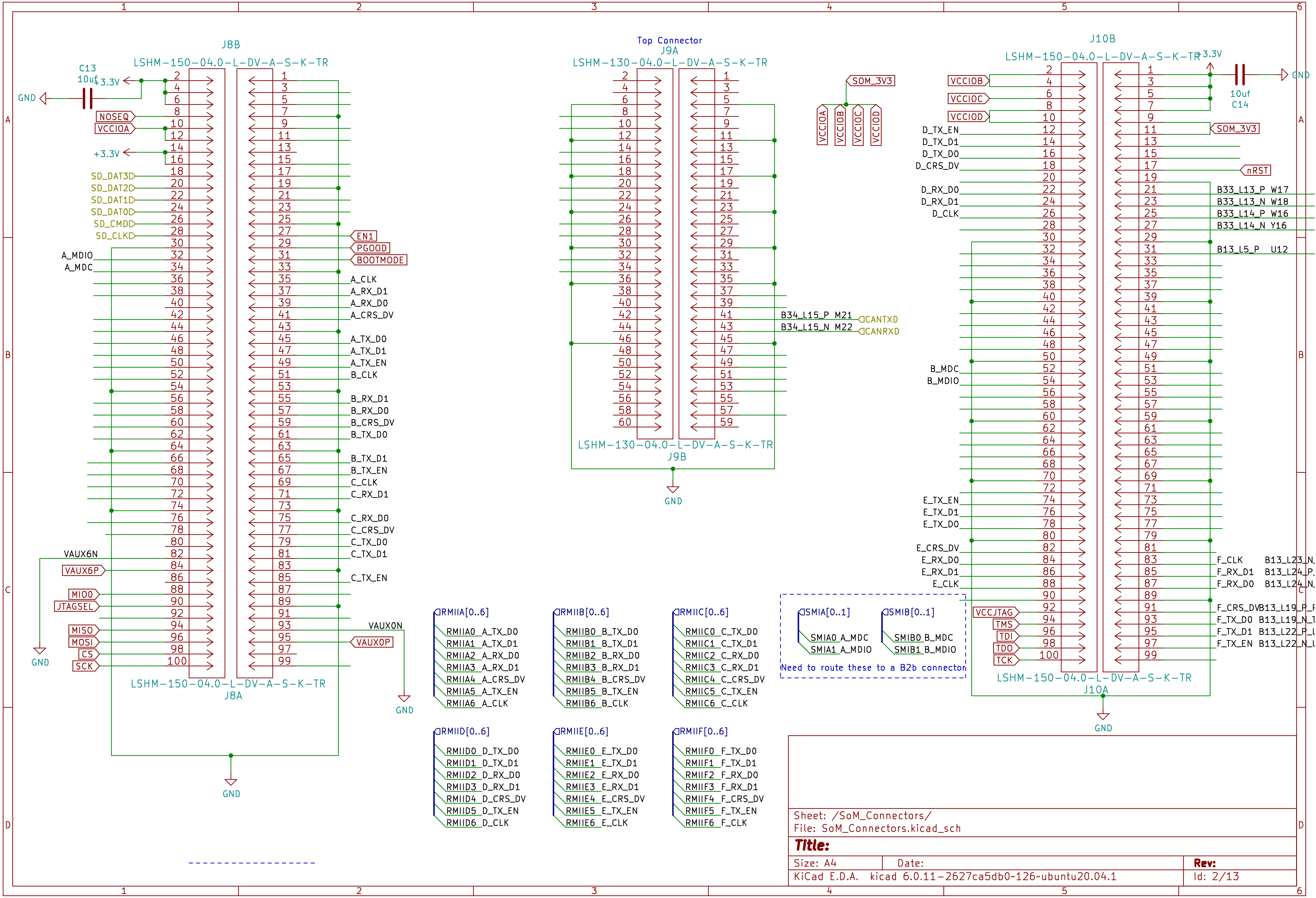
[105] "Unity simulation platform." https://unity.com/solutions/automotive-transportation-manufacturing/robotics. Accessed: 2023-09-19.

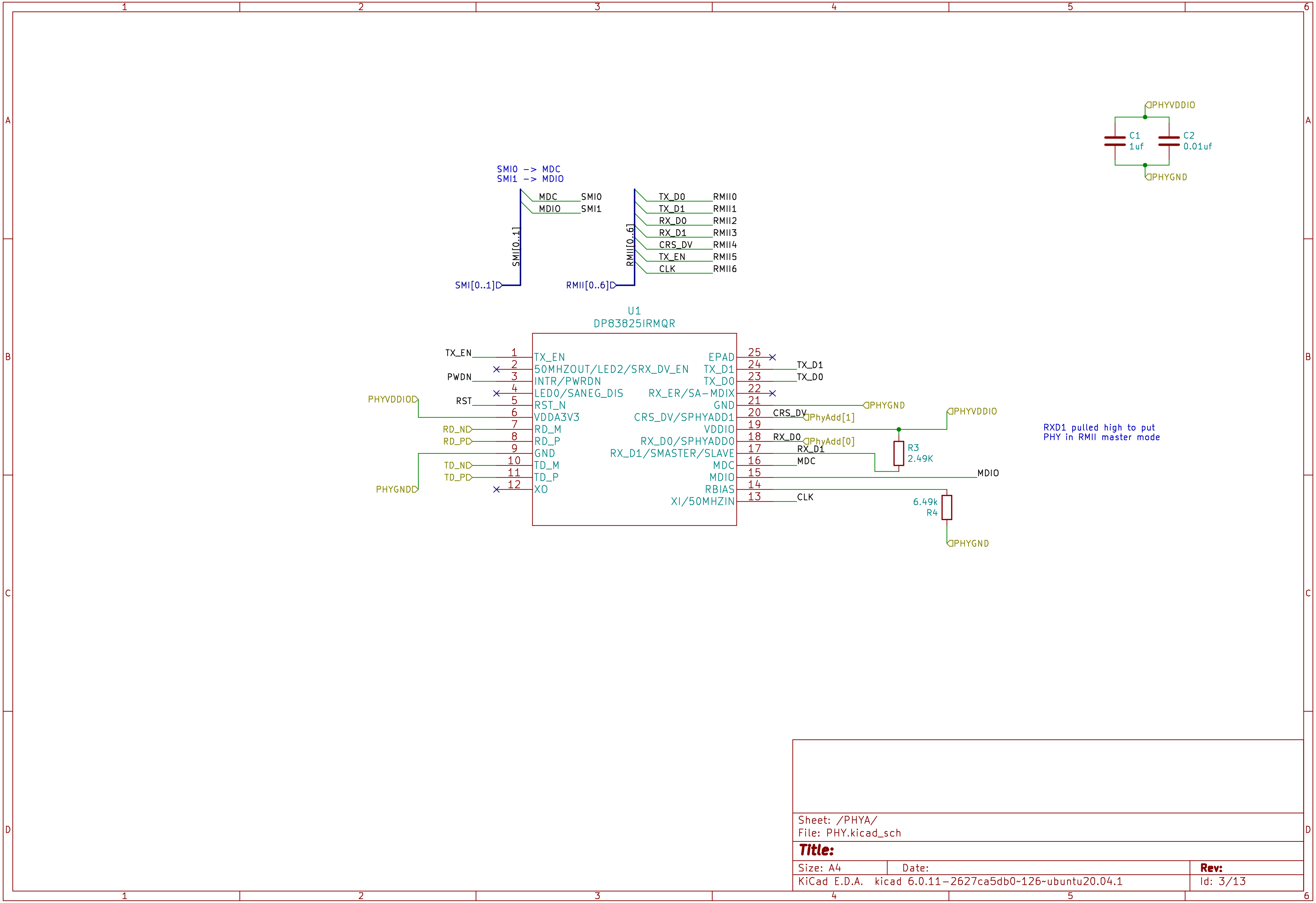
[106] "Linux sockets library." https://man7.org/linux/man-pages/man2/socket.2.html. Accessed: 2023-09-19.

Appendix A



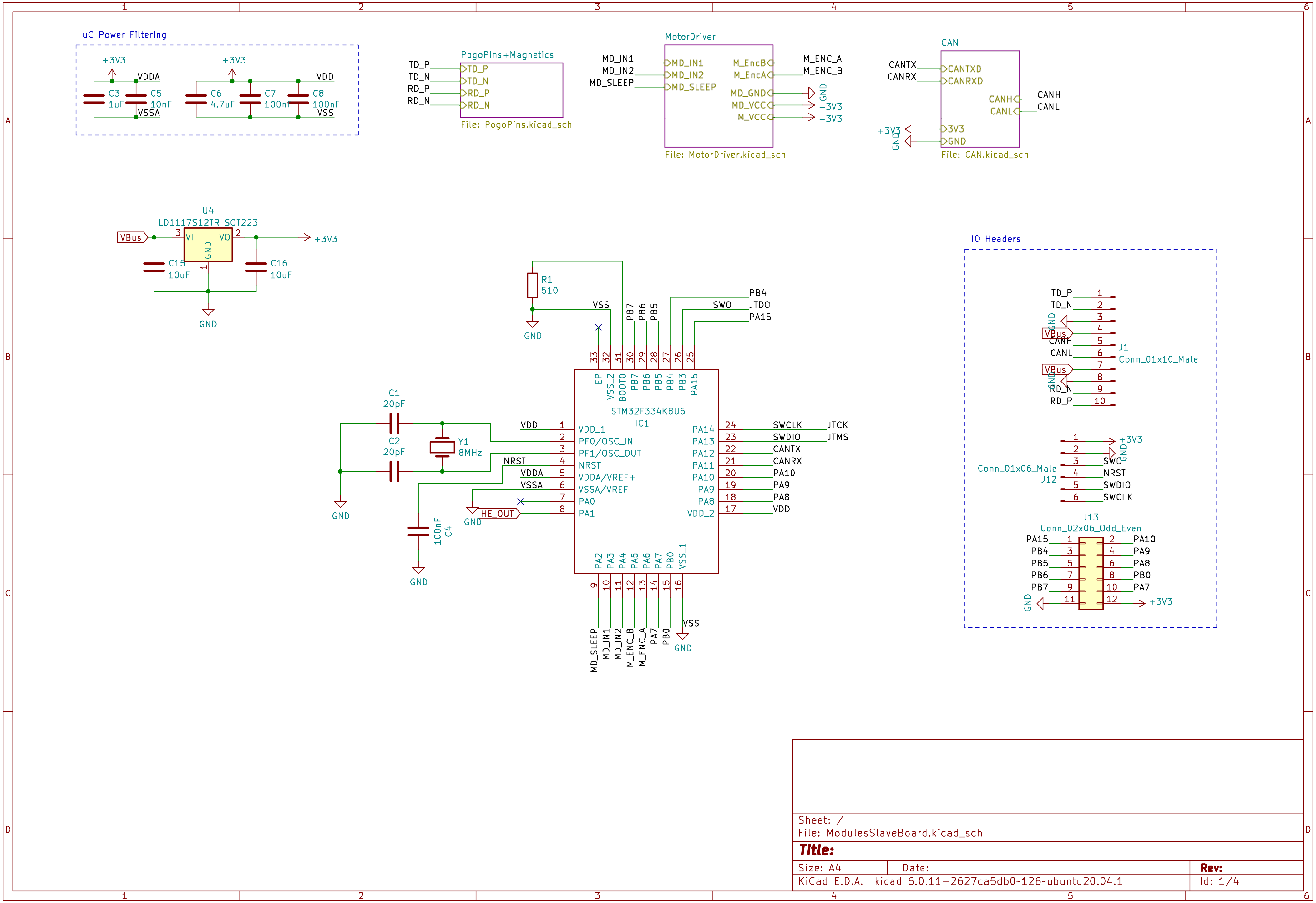
SoM_Connectors
A_CSI_D0_N
A_CSI_D0_P
A_CSI_D1_N
A_CSI_D1_P
A_CSI_CLK_N
A_CSI_CLK_P
A_CSI_GPIO0
A_CSI_GPIO1
A_CSI_SCL
A_CSI_SDA
B_CSI_D0_N
B_CSI_D0_P
B_CSI_D1_N
B_CSI_D1_P
B_CSI_CLK_N
B_CSI_CLK_P
B_CSI_GPIO0
B_CSI_GPIO1
B_CSI_SCL
B_CSI_SDA
SD_DAT3
SD_DAT2
SD_DAT1
SD_DAT0
SD_CMD
SD_CLK
MIO0
A_CLK
B_CLK
C_CLK
D_CLK
E_CLK
F_CLK
RMIIA[0..6]
RMIIB[0..6]
RMIIC[0..6]
RMIID[0..6]
RMIIE[0..6]
RMIIF[0..6]
SMIA[0..1]
SMIB[0..1]
MC_A
MC_B
MC_SLEEP
M_ENC_A
M_ENC_B
CANTXD
CANRXD
File: SoM_Connectors.kicad_sch
PHYs
SOM_3V3
PHYA
PHYB
PHYC
PHYD
PHYE
PHYF
PHYVDDIO
PHYGND
SMI[0..1]
RMII[0..6]
PhyAdd[0]
PhyAdd[1]
RD_N
RD_P
TD_N
TD_P
File: PHY.kicad_sch
2.49K
R27
R28
R29
R30
R31
R32
R33
R34
R35
R36
R37
R38
GND
Intra Tile Connectors
J1
Conn_01x10_Male
J2
Conn_01x10_Male
J3
Conn_01x10_Male
J4
Conn_01x10_Male
J5
Conn_01x10_Male
ExtVIN
CANH
CANL
Magnetics+PogoPins
File: PogoPins.kicad_sch
SD_Card
+3V3
File: SD_CARD.kicad_sch
SOM_Config
File: SOM_CONFIG.kicad_sch
PowerSupply
File: PowerSupply.kicad_sch
R1
R2
SMIA1
SMIB1
MotorDriver.sch
MD_IN1
MD_IN2
MD_SLEEP
M_EncA
M_EncB
M_VCC
MD_VCC
MD_GND
File: MotorDriver.kicad_sch
Sheet: /
File: Module Master_Board.kicad_sch
Title:
Size: A4
Date:
Rev:
KiCad E.D.A. kicad 6.0.11-2627ca5db0~ubuntu20.04.1
Id: 1/13


J8B
LSHM-150-04.0-L-DV-A-S-K-TR
C13
10uf
+3.3V
GND
NOSEQ
VCCIOA
SD_DAT3
SD_DAT2
SD_DAT1
SD_DAT0
SD_CMD
SD_CLK
A_MDIO
A_MDC
EN1
PGOOD
BOOTMODE
A_CLK
A_RX_D1
A_RX_D0
A_CRS_DV
A_TX_D0
A_TX_D1
A_TX_EN
B_CLK
B_RX_D1
B_RX_D0
B_CRS_DV
B_TX_D0
B_TX_D1
B_TX_EN
C_CLK
C_RX_D1
C_RX_D0
C_CRS_DV
C_TX_D0
C_TX_D1
C_TX_EN
VAUX6N
VAUX6P
MIO0
JTAGSEL
MISO
MOSI
CS
SCK
VAUX0N
VAUX0P
LSHM-150-04.0-L-DV-A-S-K-TR
J8A
Top Connector
J9A
LSHM-130-04.0-L-DV-A-S-K-TR
B34_L15_P M21
B34_L15_N M22
CANTXD
CANRXD
LSHM-130-04.0-L-DV-A-S-K-TR
J9B
SOM_3V3
VCCIOA
VCCIOB
VCCIOC
VCCIOD
J10B
LSHM-150-04.0-L-DV-A-S-K-TR
10uf
C14
D_TX_EN
D_TX_D1
D_TX_D0
D_CRS_DV
D_RX_D0
D_RX_D1
D_CLK
nRST
B33_L13_P W17
B33_L13_N W18
B33_L14_P W16
B33_L14_N Y16
B13_L5_P U12
B_MDC
B_MDIO
E_TX_EN
E_TX_D1
E_TX_D0
E_CRS_DV
E_RX_D0
E_RX_D1
E_CLK
F_CLK
F_RX_D1
F_RX_D0
F_CRS_DV
F_TX_D0
F_TX_D1
F_TX_EN
VCCJTAG
TMS
TDI
TDO
TCK
LSHM-150-04.0-L-DV-A-S-K-TR
J10A
RMIIA[0..6]
RMIIA0 A_TX_D0
RMIIA1 A_TX_D1
RMIIA2 A_RX_D0
RMIIA3 A_RX_D1
RMIIA4 A_CRS_DV
RMIIA5 A_TX_EN
RMIIA6 A_CLK
RMIIB[0..6]
RMIIB0 B_TX_D0
RMIIB1 B_TX_D1
RMIIB2 B_RX_D0
RMIIB3 B_RX_D1
RMIIB4 B_CRS_DV
RMIIB5 B_TX_EN
RMIIB6 B_CLK
RMIIC[0..6]
RMIIC0 C_TX_D0
RMIIC1 C_TX_D1
RMIIC2 C_RX_D0
RMIIC3 C_RX_D1
RMIIC4 C_CRS_DV
RMIIC5 C_TX_EN
RMIIC6 C_CLK
SMIA[0..1]
SMIA0 A_MDC
SMIA1 A_MDIO
SMIB[0..1]
SMIB0 B_MDC
SMIB1 B_MDIO
Need to route these to a B2b connector
RMIID[0..6]
RMIID0 D_TX_D0
RMIID1 D_TX_D1
RMIID2 D_RX_D0
RMIID3 D_RX_D1
RMIID4 D_CRS_DV
RMIID5 D_TX_EN
RMIID6 D_CLK
RMIIE[0..6]
RMIIE0 E_TX_D0
RMIIE1 E_TX_D1
RMIIE2 E_RX_D0
RMIIE3 E_RX_D1
RMIIE4 E_CRS_DV
RMIIE5 E_TX_EN
RMIIE6 E_CLK
RMIIF[0..6]
RMIIF0 F_TX_D0
RMIIF1 F_TX_D1
RMIIF2 F_RX_D0
RMIIF3 F_RX_D1
RMIIF4 F_CRS_DV
RMIIF5 F_TX_EN
RMIIF6 F_CLK
Sheet: /SoM_Connectors/
File: SoM_Connectors.kicad_sch
Title:
Size: A4
Date:
Rev:
KiCad E.D.A. kicad 6.0.11-2627ca5db0-126~ubuntu20.04.1
Id: 2/13

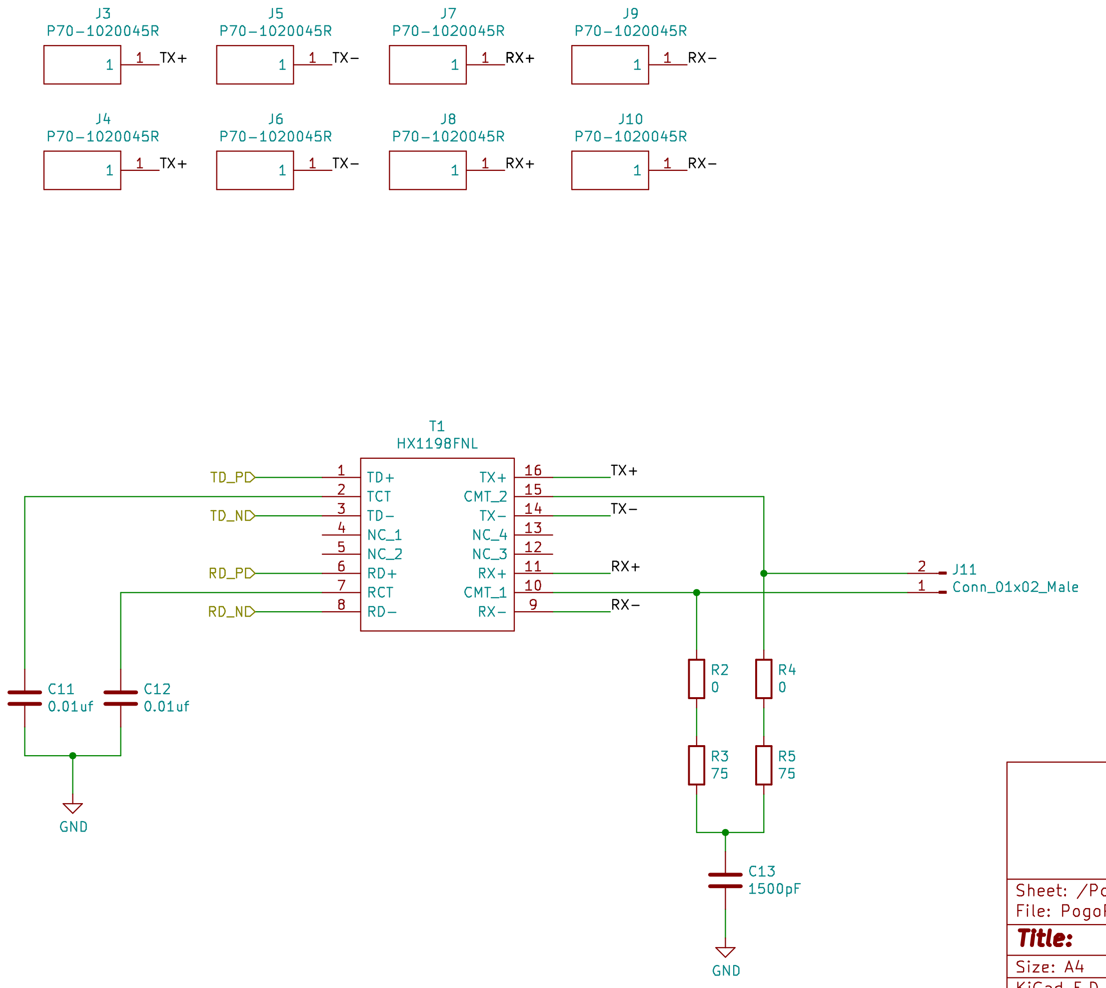


SMI0 -> MDC
SMI1 -> MDIO
MDC SMI0
MDIO SMI1
SMI[0..1]
TX_D0 RMII0
TX_D1 RMII1
RX_D0 RMII2
RX_D1 RMII3
CRS_DV RMII4
TX_EN RMII5
CLK RMII6
RMII[0..6]
PHYVDDIO
C1 1uf
C2 0.01uf
PHYGND
U1
DP83825IRMQR
TX_EN 1 TX_EN
2 50MHZOUT/LED2/SRX_DV_EN
PWDN 3 INTR/PWRDN
4 LED0/SANEG_DIS
RST 5 RST_N
PHYVDDIO 6 VDDA3V3
RD_N 7 RD_M
RD_P 8 RD_P
9 GND
TD_N 10 TD_M
TD_P 11 TD_P
PHYGND 12 XO
EPAD 25
TX_D1 24 TX_D1
TX_D0 23 TX_D0
RX_ER/SA-MDIX 22
GND 21 PHYGND
CRS_DV/SPHYADD1 20 CRS_DV PhyAdd[1]
VDDIO 19 PHYVDDIO
RX_D0/SPHYADD0 18 RX_D0 PhyAdd[0]
RX_D1/SMASTER/SLAVE 17 RX_D1
MDC 16 MDC
MDIO 15 MDIO
RBIAS 14
XI/50MHZIN 13 CLK
R3 2.49K
6.49k R4
PHYGND
RXD1 pulled high to put
PHY in RMII master mode
Sheet: /PHYA/
File: PHY.kicad_sch
Title:
Size: A4
Date:
Rev:
KiCad E.D.A. kicad 6.0.11-2627ca5db0~126~ubuntu20.04.1
Id: 3/13

Appendix B



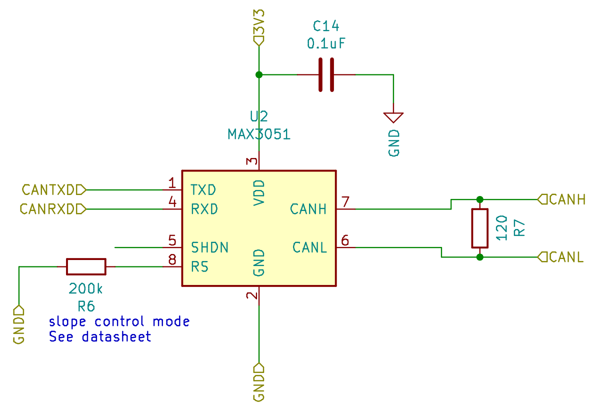
uC Power Filtering
+3V3
VDDA
C3 1uF
C5 10nF
VSSA
C6 4.7uF
C7 100nF
C8 100nF
VDD
VSS
PogoPins+Magnetics
TD_P
TD_N
RD_P
RD_N
File: PogoPins.kicad_sch
MotorDriver
MD_IN1
MD_IN2
MD_SLEEP
M_EncB
M_EncA
MD_GND
MD_VCC
M_VCC
M_ENC_A
M_ENC_B
GND
File: MotorDriver.kicad_sch
CAN
CANTX
CANRX
CANTXD
CANRXD
CANH
CANL
3V3
GND
File: CAN.kicad_sch
U4
LD1117S12TR_SOT223
VBus
VI
VO
C15 10uF
C16 10uF
R1 510
VSS
PB4
SWO
JTDO
PA15
PB7
PB6
PB5
EP
VSS_2
BOOT0
PB3
STM32F334K8U6
IC1
C1 20pF
C2 20pF
Y1 8MHz
VDD_1
PF0/OSC_IN
PF1/OSC_OUT
NRST
VDDA/VREF+
VSSA/VREF−
PA0
PA1
HE_OUT
C4 100nF
PA14
PA13
PA12
PA11
PA10
PA9
PA8
VDD_2
SWCLK
JTCK
SWDIO
JTMS
PA2
PA3
PA4
PA5
PA6
PA7
PB0
VSS_1
MD_SLEEP
MD_IN1
MD_IN2
M_ENC_B
M_ENC_A
IO Headers
J1
Conn_01x10_Male
Conn_01x06_Male
J12
J13
Conn_02x06_Odd_Even
Sheet: /
File: ModulesSlaveBoard.kicad_sch
Title:
Size: A4
Date:
Rev:
KiCad E.D.A. kicad 6.0.11-2627ca5db0~126~ubuntu20.04.1
Id: 1/4



Sheet: /PogoPins+Magnetics/
File: PogoPins.kicad_sch

Title:

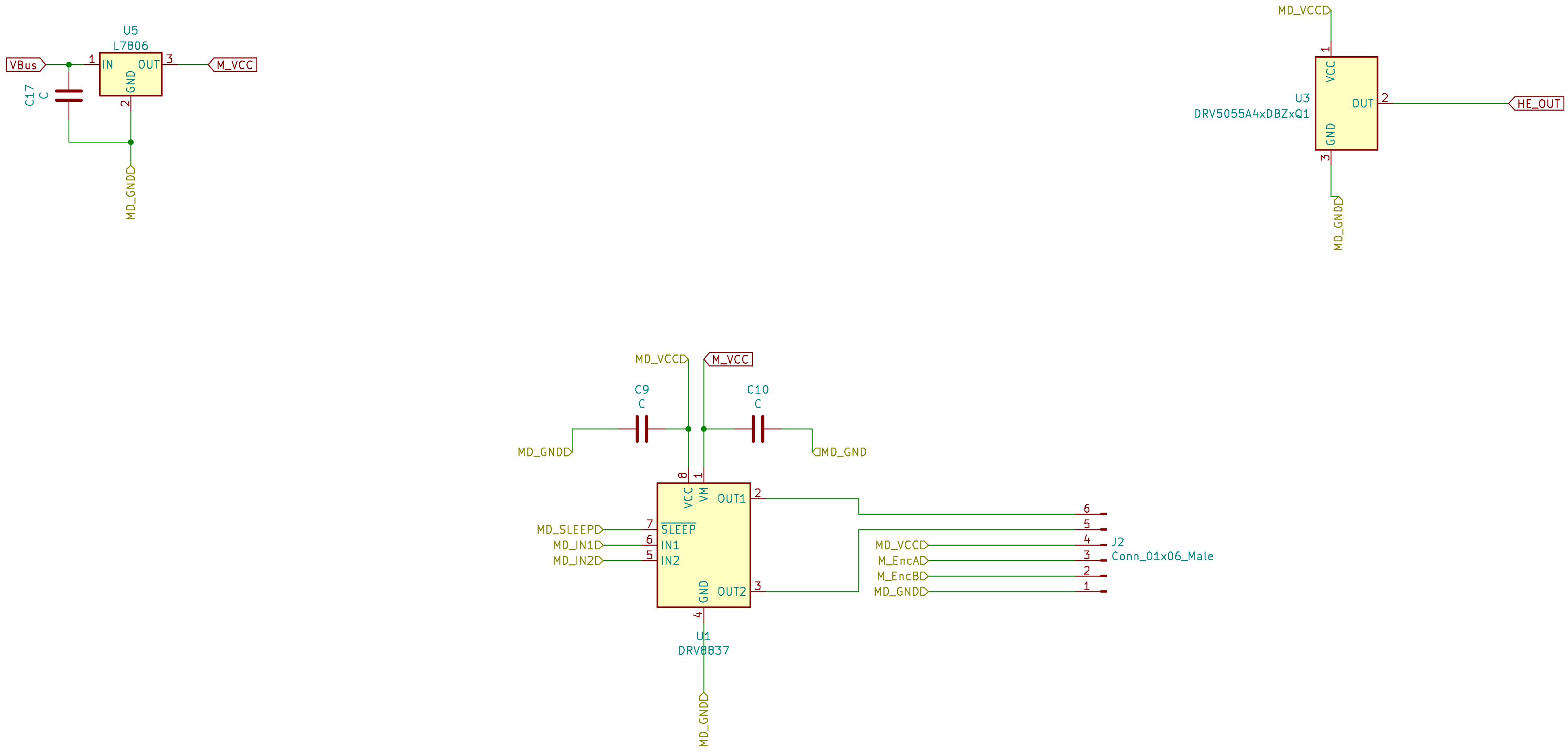
Size: A4	Date:	**Rev:**
KiCad E.D.A. kicad 6.0.11-2627ca5db0~126~ubuntu20.04.1		Id: 2/4



Sheet: /CAN/
File: CAN.kicad_sch

Title:

Size: A4	Date:	**Rev:**
KiCad E.D.A. kicad 6.0.11-2627ca5db0~126~ubuntu20.04.1		Id: 3/4



Sheet: /MotorDriver/
File: MotorDriver.kicad_sch

Title:

Size: A4	Date:	**Rev:**
KiCad E.D.A. kicad 6.0.11-2627ca5db0~126~ubuntu20.04.1		Id: 4/4