Symbiotic Multi-Agent Construction 3.0

A Major Qualifying Project Report submitted to the faculty of
WORCESTER POLYTECHNIC INSTITUTE
in partial fulfillment of the requirements for the degree of Bachelor of Science

Submitted By:

Jacob Leavitt
Computer Science

Kayla Lepping
Robotics Engineering

Jose Rivera
Mechanical Engineering

Brian Shin
Mechanical Engineering, Robotics Engineering

27 April 2023

Professor Greg Lewin, Advisor
Robotics Engineering Department

Professor Carlo Pinciroli, Advisor
Robotics Engineering Department

Abstract

Swarm robotic solutions have the potential to solve many of the dangers and issues current construction practices have today. By having many robots working on a single project, construction can be completed in a fast and efficient manner. The overall intent for this project is to design a versatile robotic swarm system capable of autonomously constructing complex structures by means of inchworm-like linkage robots and standardized interconnected "smart" blocks. This project introduces a new backbone linkage system in the blocks and improved stigmergic control that builds upon the groundwork established by past iterations of the project - enabling the construction of higher-complexity structures. Powering the system is a complete algorithm capable of detecting if any structure is stable during and after construction while building the blueprints to be propagated among the blocks.

Acknowledgments

We would like to thank our advisors: Professors Carlo Pinciroli and Greg Lewin for their support and insights during the development of this project. We would also like to thank our predecessor MQP teams: SMAC 1.0, MARIA, and RAINSTORM for laying down the groundwork for this project.

Authorship

Report Section	**Author(s)**
Front Matter	Jacob Leavitt, Kayla Lepping, Jose Rivera, Brian Shin
1: Introduction	Jacob Leavitt, Brian Shin
2: Related Work	Jacob Leavitt, Kayla Lepping, Jose Rivera, Brian Shin
3: Design and Development	Jacob Leavitt, Jose Rivera, Brian Shin
4: Results and Analysis	Jacob Leavitt, Brian Shin
5: Conclusions	Jacob Leavitt, Jose Rivera, Brian Shin

Project Work	**Author(s)**
Mechanical Design	Jose Rivera, Brian Shin
Electrical Design	Brian Shin
Embedded Software	Brian Shin
Algorithm Design & Implementation	Jacob Leavitt
Manufacturing	Jose Rivera, Brian Shin

Table of Contents

List of Figures

List of Tables

Executive Summary

The Symbiotic Multi-Agent Construction (SMAC) 3.0 project is the fourth iteration of the 3D Swarm Construction project, following the RAINSTORM Project [9,10]. SMAC 3.0 redesigned the building material first developed by M.A.R.I.A as part of an autonomous building platform using inchworm building agents and smart blocks [7]. Developed alongside the new block design is a blueprint generation algorithm that validates a structure under weight and stability constraints defined by the inchworm and new block capabilities and generates instructions for any buildable structure.

Following previous iterations and related work in the autonomous construction field, this project aimed to strengthen the building material by creating a new peg design, increasing the shear and tensile strength of connections between blocks and the building agent. A new IR communication system was implemented over the previous visual light system for increased transmission reliability. The new blueprint algorithm was developed as an easy to use structure creation simulation followed by full system validation and blueprint generation.

In order to achieve a versatile and stable system that would be able to build structures that were not possible in the previous iteration of the project, our team had to significantly increase the shear strength of the connections between individual blocks. Since the blocks have limited space within the shell, the team decided to create a non-actuated linkage mechanism, because an actuated mechanism would require internal components that would take up too much space along with the communication components. With these constraints in mind, our team decided to implement a peg linkage system which would allow each block to perform strong-links and weak-links depending on the needs of the structure being built. This idea was developed into the new backbone linkage system, which allowed structures to have internal backbone-like supports that could change directions when needed using a block with a 90 degree strong-link configuration. The team made prototypes with both permanent and removable pegs, then tested these design groups to evaluate their strengths and weaknesses. Static force tests yielded similar results between both design groups, however the removable peg prototype was the more versatile of the two during structure testing. The removable pegs allowed weak-links on the faces that support strong-links when a peg was not inserted, which was necessary for some of the sub-structures created during these tests. Furthermore, building with the permanent peg prototypes would complicate the necessary movements of the construction agent more than its removable counterpart. Force testing shows improved loading capability before separation with another strong-linked block, over previous iteration (MARIA MQP) by 53% in tensile loading and 330% in shear loading. These improvements provided the overall system with the ability to create stable structures that were previously impossible in the past iterations of SMAC.

The communication system was developed to establish reliable communication between the blocks for block-to-block and block-to-robot communication. We aimed to demonstrate scalability, low error, and low transmission time. Past iterations of the SMAC platform used near-field communication (NFC) or visual light communication (VLC) for information exchange between blocks. NFC proved effective in the SMAC 1.0 project for communication between agents, while MARIA's VLC system offered a novel means of visual indication and communication between agents; however, it suffered from reliability issues with color sampling. Our team chose to use infrared (IR) communication to evaluate a lightweight and robust method of communication, since it continuously proves its reliability by its widespread use in modern consumer electronics. From developmental unit tests of varying distance and translucent obstructions between a transmitter and a receiver, we found that IR communication was cheap, reliable, and robust.

The blueprint generation algorithm was developed to allow custom structure creation in a simulation, then develop a set of instructions for the smart blocks and inchworms to follow. By running a single program, a user can build a single structure in a simulation where it's processed by the algorithm to determine the orientation and placement of the new peg design and output a blueprint if the structure is buildable. The algorithm guarantees that a blueprint will only be generated if the generated peg placement is buildable by the inchworm under the weight and stability constraints of the system.

In a 10x10x10 3D grid, any structure that is built can be processed with a time complexity of <500ms. The algorithm is capable of processing hundreds of blocks at once, validating the placement and stability of each one as it simulates construction placement block by block. Using a prioritization heuristic for peg placement, the algorithm chooses to link to blocks based on their weight value and block classification. The algorithm guarantees that if a solution is found that it can be built by the system. This is to optimize the time complexity when checking multiple orientations on structures of hundreds of blocks proved much slower.

The new block design and IR communication protocol improved the stability and reliability of the SMAC construction system, allowing building capabilities including overhangs of 3 blocks able to support an inchworm and data transmission rates peaking between 14.8 and 22.2 bytes/s. The blueprint generation algorithm is more complete than any other iteration taking into account the heterogeneous block design, a user friendly simulation, weight and stability validation, and blueprint generation. It is also flexible in its constraints for maximum weight or maximum overhang limits for the block design. SMAC 3.0 has laid the groundwork for a stronger autonomous construction platform using an inchworm and smart block design.

Chapter 1: Introduction

1.1 Background

Robotics in construction is not a new phenomenon. Drones have been helping construction crews survey sites for years to limit risk to humans. New heavy machinery comes with assistive technology to help the machine adapt to environmental conditions [1]. A new company called Contour Crafting has recently created house-sized robotic 3D printers which can efficiently build single-family houses in under 24 hours [2]. However, the robotic construction of houses still requires monitoring by humans and a centralized system manager.

Swarm robotic solutions have the potential to enable construction in places previously impossible for humans. By having a distributed autonomous system, robots can build entire structures without the need to communicate with humans. With no single point of failure in a system and no need for constant communication, robots could be sent to places such as the Moon or Mars to build structures before humans arrive. With a scalable swarm of robots, the ability to create larger structures within a specified time frame increases. Decentralization of a main controller also allows the swarm to have a margin of failure and allows for easy replacement if any robot fails.

Where typical construction involves a site manager as a central point of intelligence who directs other workers to construct a building, this project aims to decentralize the planning of a building to a smart-construction medium that the system utilizes. This phenomenon is known as *stigmergy*, which involves the use of indirect communication through the environment to coordinate the actions of individuals towards a common goal. Ants are an example of stigmergic builders with no centralized source of information, which all construction workers utilize in a construction environment [3]. In stigmergy, each individual modifies its environment by leaving a trace of its activity, which serves as a stimulus for the next individual to continue the task. For example, ants might lay down a chemical trail that directs other ants to food sources, but in a decentralized construction system, the active environment stores and relays data to the building agents so that they know where and how the next smart-construction medium must be placed. Taking this as inspiration, the Symbiotic Multi-Agent Construction (SMAC) platform has established smart modular building blocks as the construction medium, which will be able to guide construction agents to the next location to place a block without a need for a central controller.

The benefits of an autonomous construction system allow for applications that are typically too dangerous or difficult. In the future, SMAC can be applied to projects at the deepest parts of the sea, or on other planets. SMAC's inchworm builder robot's ability to cross 3-dimensional terrain creates many new opportunities for construction in previously unreachable locations. One advantage of an autonomous construction system that builds structures with intelligent blocks is that it could allow for the continuous monitoring of the entire building with sensors built into the blocks themselves. In deep sea or space applications where structural integrity is essential, these blocks could self-report damages due to pressure changes or broken seals. Ultimately, the SMAC project aims to implement a heterogeneous block building material with increased connection strength between faces.

1.2 Problem Statement

The overall goal of this project is to design a versatile robotic swarm system capable of autonomously constructing complex structures by means of inchworm-like robots and interactive "smart" construction materials. This project builds upon the groundwork established by the previous versions of SMAC. In these previous iterations, smart construction materials in the form of homogeneous cubes or "blocks" have introduced allen key screws or permanent magnets on a face as means of mechanically mating these blocks [7,9]. SMAC 3.0 aims to strengthen the connection between blocks to support higher loading forces and create stable complex structures more representative of real-world use. To achieve stigmergic behavior in the robotic swarm construction system, the team must design a novel blueprint construction algorithm that is scalable and decentralized to work with the new smart blocks and validate an input structure for weight and stability constraints.

1.3 Statement of Contribution

The SMAC 3.0 team designed a new means of mechanical mating between blocks necessary for the creation of stable structures, involving a heterogeneous block design and a novel backbone linkage system to increase overall strength and stability in the structures that are built. These mechanical improvements allow for the construction of non-trivial structures (e.g., large structures with significant strength and resemblance to real architectural features). By increasing the strength of the inter-block connections in identified substructures necessary for most construction applications, we expand the system's capability in the scale, complexity, and loading applications for the structures it creates.

Further, the system for block-to-block and block-to-robot communication was made more reliable and robust with the use of infrared (IR) light signals. Infrared light transmission is a widely used communication protocol for transmitting data found in many modern consumer electronic devices. With a medium of distributed information exchange established between agents of construction, we present a novel and decentralized blueprint generation algorithm. This algorithm verifies the structural stability of a given structure and solves for placement priority and strong-link connections based on real-world constraints. If the given structure passes our validation, a blueprint containing necessary information for stigmergic building is generated to be propagated amongst the smart blocks. Ultimately, our contributions to the SMAC platform show that a heterogeneous block system is feasible and effective in creating stable structures that are representative of the substructures of a larger building.

1.4 Design Novelty

SMAC 3.0, while rooted in three years of prior work, addresses and improves upon several of the limitations of its predecessors. This project modified the smart blocks from the MARIA iteration to establish a new method of communication between blocks that has not been evaluated before in past iterations. The team also developed a novel backbone system integrated with the smart blocks, involving removable pegs to support higher loading forces, which enabled the creation of structures more complex than what was possible in past versions of the project. The new block mating system creates two distinct connections: strong links and weak links, and constrains the strong links to only occur between blocks at 90° and 180° angles - since each block can support up to two strong links. In previous contributions to SMAC, homogeneous smart block designs were explored to prioritize either strength or orientation independence; however, neither of these approaches were "good enough" by our definitions. We developed a heterogenous block prototype to work towards finding a balanced compromise of strength and algorithmic complexity.

The behaviors of both the robots and the blocks are dictated by a novel and decentralized construction algorithm. While the introduction of different types of nodes adds a level of algorithmic complexity, the innovative controls in SMAC 3.0 make for a robust and stigmergic construction process. Overall, the SMAC 3.0 platform is capable of building strong and complex structures which better reflect real-life constraints of construction.

Chapter 2: Related Work

This chapter presents and discusses research relevant to smart structures, robotic swarm construction platforms, and self-reconfigurable swarms.

2.1 Robotic Swarm Construction

Nagpal et al. [4] developed an autonomous robotic system for three-dimensional collective construction (TERMES) that used small mobile builder robots paired with passive specialized blocks to build predetermined structures. Their system allowed the builder robots to travel on and off the frame and build large structures. TERMES' centralized algorithm generated blueprints for construction and its decentralized execution of these blueprints allowed multiple robots to build simultaneously without collision. An image of a real builder robot on the blocks and a simulation showing multiple robots building a structure is shown in Figure 1.

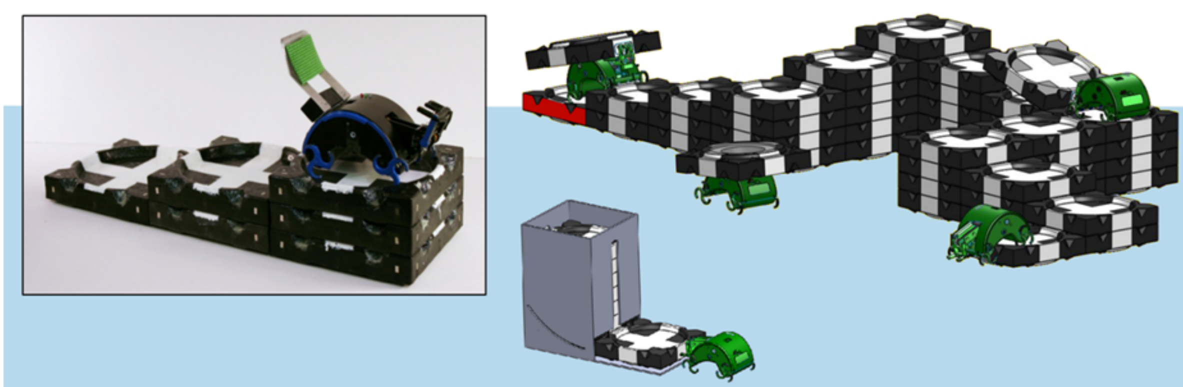


Figure 1. A small TERMES structure and robot (left) and simulated construction of a larger structure by multiple robots (right).

The TERMES platform had limitations regarding the types of structures that could be built. Since the building blocks had no means of mechanical mating between the sides of adjacent blocks, structures with overhanging features could not be created. The builder robots also climbed using "wegs," a combination of wheels and legs, which only allowed them to climb small vertical distances. This limitation meant that a stair system must be built for the robots to create taller structures, using excess materials and heavily limiting potential structures. Keeping this in mind, the SMAC platform, both past and current iterations, utilizes 5-DOF inchworm robots that can travel across greater heights and building blocks that can connect in all directions, allowing for overhangs and other complex structures.

The SMAC platform's bipedal inchworm robot was largely influenced by the Bipedal Isotropic Lattice Locomoting Explorer (BILL-E) system proposed by Jennet and Cheung [5]. BILL-E is a robotic swarm construction system inspired by multi-degree of freedom (DOF) industrial robots that are fixed to gantry systems to enable larger building areas. The system can lay out structures composed of cubic lattice blocks seen in Figure 2 and uses a multi-DOF robot arm with grippers to attach to its environment and building materials.

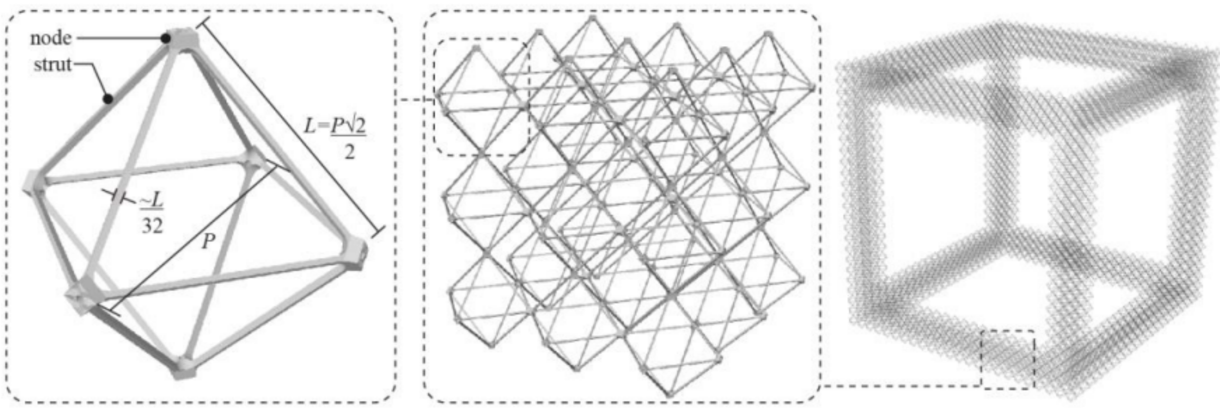


Figure 2. A cubic lattice structure composition and geometry

The robot was designed to move in its three-dimensional isotropic lattice environment, which is fully composed of the building blocks. The functional requirements for operation are listed:

1. The robot should be able to traverse linearly (X, Y, or Z)
2. The robot should be able to turn and traverse in the direction orthogonal to the first direction (X to Y, Y to Z, X to Z)
3. The robot should be able to turn up/down concave and convex corners
4. The robot should be able to step up/down a level (+/- Z)

To optimize effective navigation on three-dimensional structures, the grippers include fault-tolerant end effectors to maximize the probability of success in navigation. Through a working prototype, the BILL-E system demonstrated the ability to effectively satisfy its navigation requirements and hold and place lattice 9 elements, shown in Figure 3 [5].

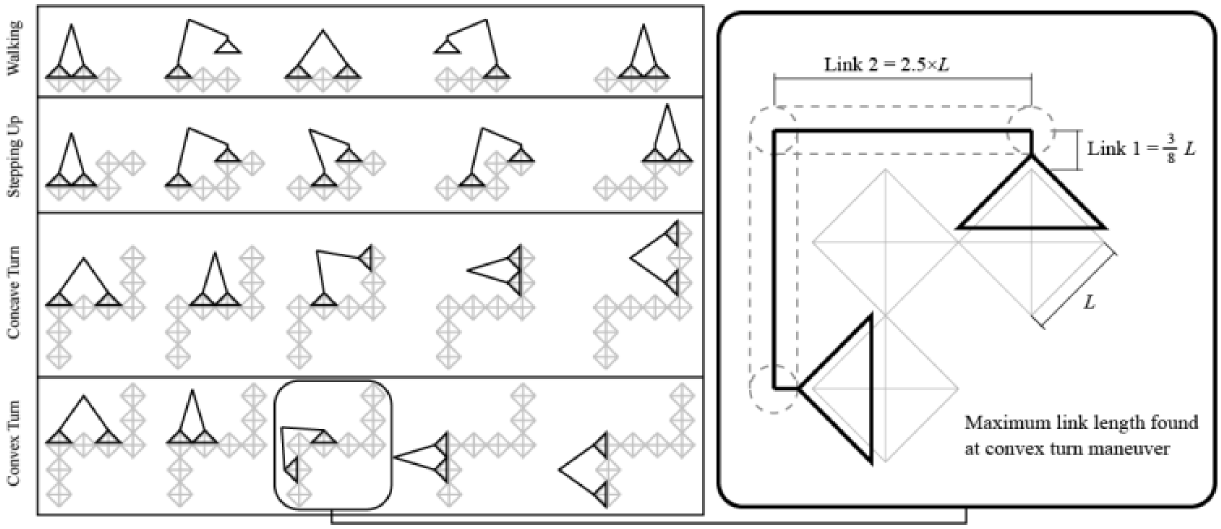


Figure 3. Robot positioning variations based on DOF functional requirements for BILL-E.

Due to the focus on the three-dimensional motion within its mobility requirements, the BILL-E robot and block design is an example of a physical implementation of a modular construction system with few restrictions on the types of structures it can produce. The system design presented in this work will take inspiration from the BILL-E robot and blocks while incorporating and optimizing the intelligent building blocks to achieve enhanced versatility in the classes of structures that our system can produce. BILL-E can move across inclined surfaces, including vertical surfaces, by gripping the structure. This way the robot does not tip over or slip off the structure when changing vertical levels of the building. This makes the bipedal inchworm structure of the BILL-E system favorable for maintaining a stable system with minimized chances of mechanical failure. SMAC incorporates the communication aspect of BILL-E of cube to cube and cube to building agent to allow more versatile structures.

2.2 SRoCS Smart Structures

One of the most efficient means of robotic construction involves the coordination and communication of multiple robots, which we call *swarms*. As a swarm, robots must work together to coordinate the building of a structure. While swarm communication is often established between robots, a different area of research incorporates robot-to-structure communication, in which the building blocks for the structures consist of smart elements. This means that robots can directly interface with the structure itself and other robots, which allows for actions and calculations to be computed externally without robots and instead be computed by the structure. Having an intelligent structure also helps to establish a communication medium where the robots can relay information to and from other robots in the swarm.

The Swarm Robotics Construction System (SRoCS) uses Near Field Communication (NFC) and computer vision to enable the communication between robots and building materials (smart blocks) seen in Figure 4 [6]. The SRoCS platform aimed to mimic stigmergic behavior - an adaptive mechanism of indirect coordination that insects like wasps, termites, and ants use to communicate tasks. This method of coordination is used in ant colonies through pheromones that signal to worker ants the tasks that need to be completed. The pheromones are left behind by other worker ants that have worked in the area [3].

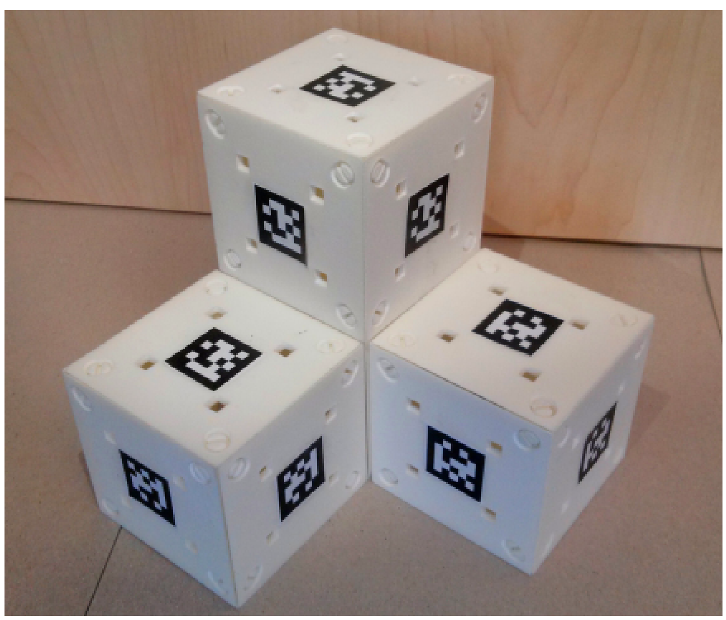


Figure 4. SRoCS smart blocks.

The two parts of the system include intelligent blocks and BeBots that pick and place these blocks. The BeBots seen in Figure 5 are not involved in decisions regarding the organization of the end structure and rely on the smart blocks interface for build instructions. Although this allowed for flexibility during construction, there was no way to fully plan or predict the final structure [6]. To assist with the localization of the robotic manipulator with respect to the building blocks, SRoCS uses AprilTag tags. These allow the robot to use computer vision to find blocks and determine their position in their environment.

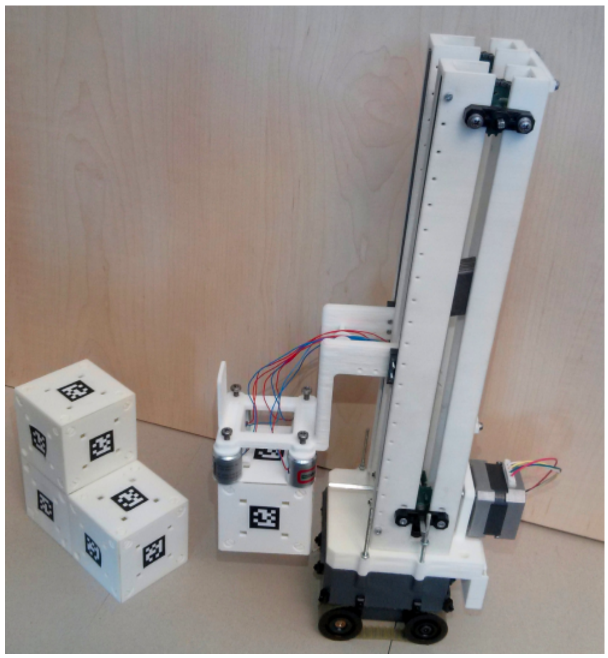


Figure 5. BeBot robot manipulation a SRoCS block.

Later work on SRoCS explored a smart material to help guide construction and convey information to the system. Each block had color LEDs and NFC modules to communicate with the BeBots. The blocks updated the LEDs on their faces according to defined rules, and those LEDs indicated to the BeBots how to continue construction. This passive transfer of information allowed for a decentralized system where construction decisions came from dynamic signals. This idea was adapted into the MARIA project [7], a SMAC adaptation where the smart blocks seen in Figure 6 were significantly improved by updating LEDs on block agent faces to indicate the next steps of construction, acting as both a messaging system and visual indicator of the state of construction.

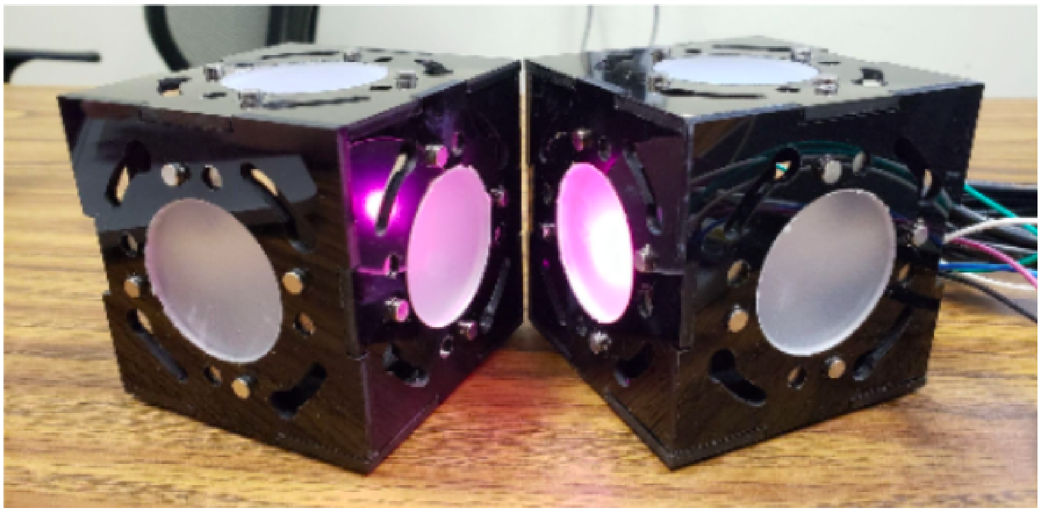


Figure 6. MARIA (SMAC 2.0) smart blocks.

The SRoCS platform is an example of how blocks that convey information about the structure allow for a more capable system. The advantage of having a structure be aware of its current configuration is that it can communicate what is missing to the worker robots.

2.3 Self-Reconfigurable Swarms

ATRON is a versatile robotic platform presented by Brandt et al. [8] that features a swarm of self-configurable modules. Each module is capable of autonomous reconfiguration to provide functionality to the system's current requirements. The platform can perform fixed-topology locomotion, cluster flow locomotion, and object manipulation.

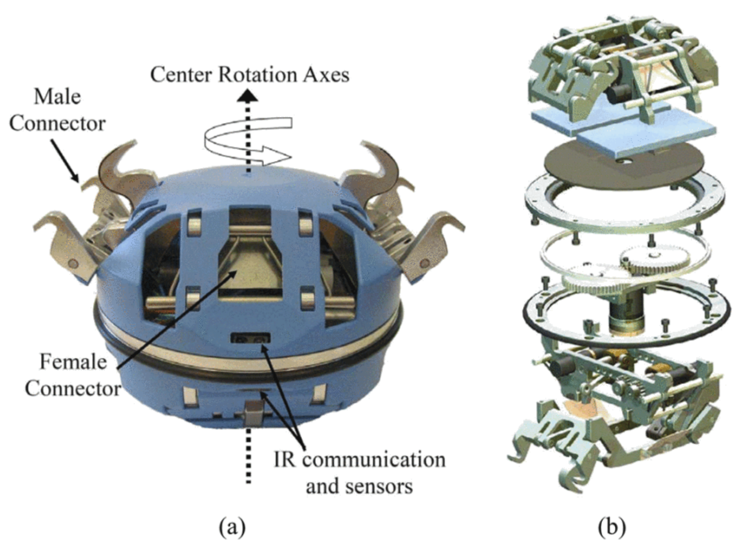


Figure 7. The ATRON self-reconfigurable robotic module and its exploded mechanical assembly.

The ATRON robotic module, as shown in Figure 7, takes a spherical form with gendered connectors that can mate up to eight neighboring modules, with four on each hemisphere. The robotic sphere actuates about its equator with continuous rotation, granting it a single degree of freedom (DOF). One module can lift up to two other connected modules against gravity. When assembled as a swarm, the modules are structured in a surface-centered cubic lattice such that the rotation axes of two connected modules are perpendicular to one another. Inter-module communication is achieved via infrared light and reflects stigmergic behaviors through rule-based control - meaning that each of the identical modules can perform actions based on their internal state machines and sensors. Other sensors on an ATRON module allow it to sense its surroundings and its tilt with respect to gravity.

The robotic swarm platform can self-configure into fixed structures that allow for locomotion, such as those that resemble a car or a snake. The researchers on this project explored several different configurations of "wheeled" vehicles, ranging from 4 to 14 wheels. From this, they found that each set of wheels adds about 25 N of drag and that the car can operate for more than two hours with a top speed of 6 cm/s. The other structures, including a snake and a cluster walker, demonstrate the system's versatility regarding operation in tight spaces and rough terrains. Moreover, ATRON researched another form of locomotion: cluster flow. With cluster flow locomotion, the modules in the swarm move from one general cluster to another (e.g. back of the swarm to the front), allowing the whole swarm to move. Compared to fixed-topology locomotion, cluster flow is slower and less efficient; however, it can adapt to the environment as needed. ATRON is also capable of object manipulation by transforming into structures such as serial linkage robotic arms or as a flat conveyor surface to move objects.

While the ATRON platform does not *build* structures like SMAC 3.0 sets out to do, there are a couple of attractive concepts that it demonstrates that may prove to be useful for SMAC. Firstly, ATRON is a successful and scalable implementation of swarm robotics that replicates stigmergic behavior through its decentralized control algorithm. Further, the mechanical mating and robust IR communication between modules reflect a similar one-to-many relationship that SMAC 3.0 aims to demonstrate with the smart blocks.

2.4 Previous Project Work

2.4.1 SMAC 1.0

SMAC 1.0 designed a multi-robot system that could build the scaffolding of larger structures [9]. The two types of robots used for construction are builder robots - bipedal inchworm robots, inspired by those from BILL-E - and smart building blocks. Figure 8 shows the design of the robots and potential configurations while traversing a structure of smart blocks. The goal of SMAC 1.0 was to create a distributed system that removed any single point of failure in the platform allowing for the autonomous construction of structures.

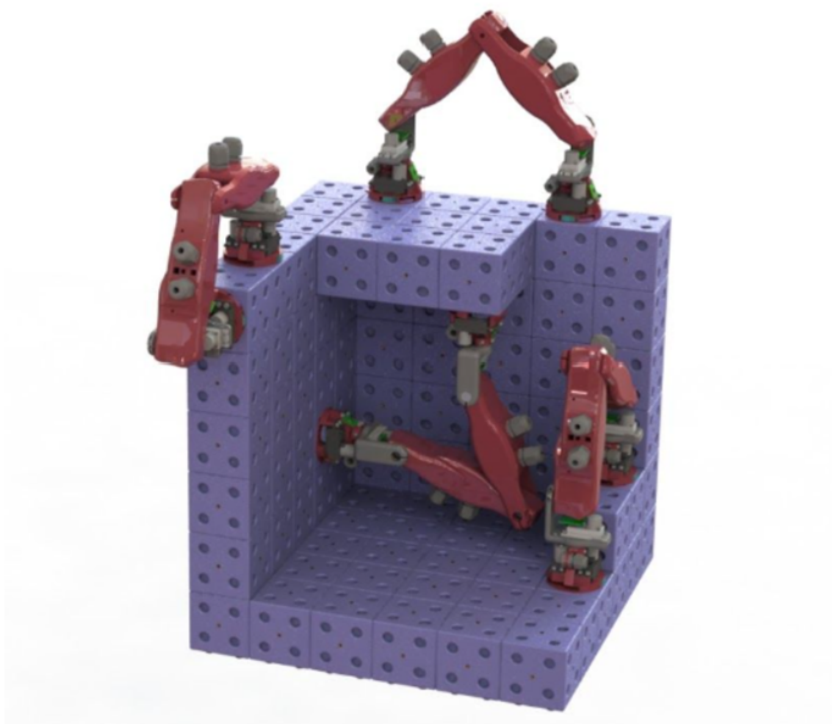


Figure 8. Simulation of multiple inchworm robots traversing multiple planes while building a structure. [9]

One of the primary focuses of SMAC was the communication between the builder(s) and the cubes. The robots send data to the structure including: where it believes it is on the structure and updates on the status of their task. The structure then checks that the robot(s) is still working, and if the signal is not received, it sends messages to the robot(s) to activate them. The cubes also could communicate their location and status and the location of each robot. Having these tasks be completed by the structure allows for less electrical complexity on the builder robot(s).

The cube design proposed by this project used an Allen key screw system that enables the inchworm to attach itself to blocks as it ferries and scales structures. The proposed block design was heterogeneous as it only had one side with a male Allen key connection, with the rest being female as seen in Figure 9.

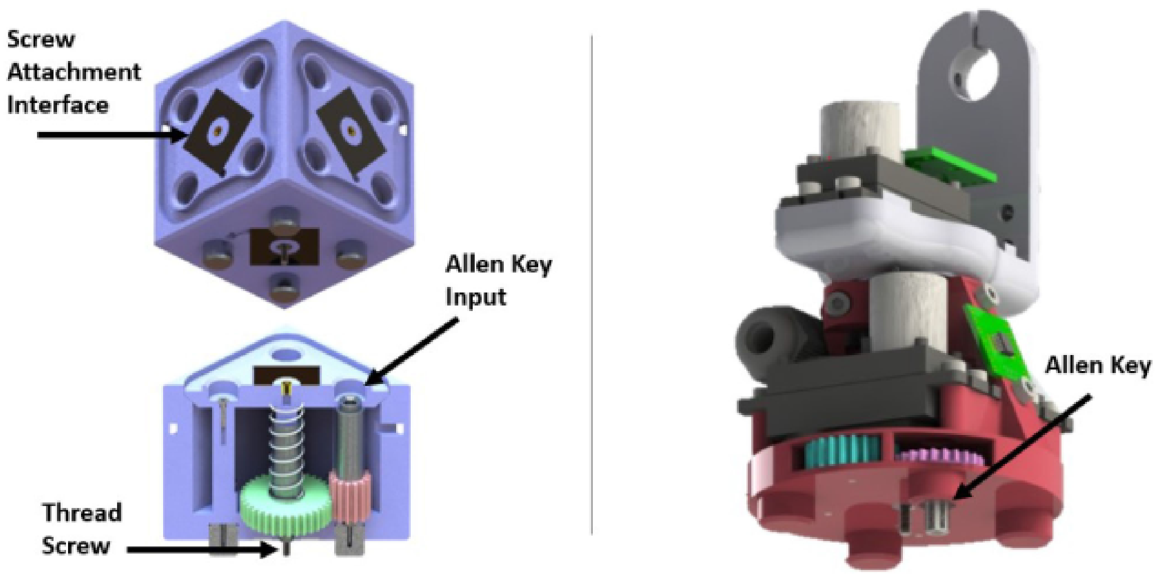


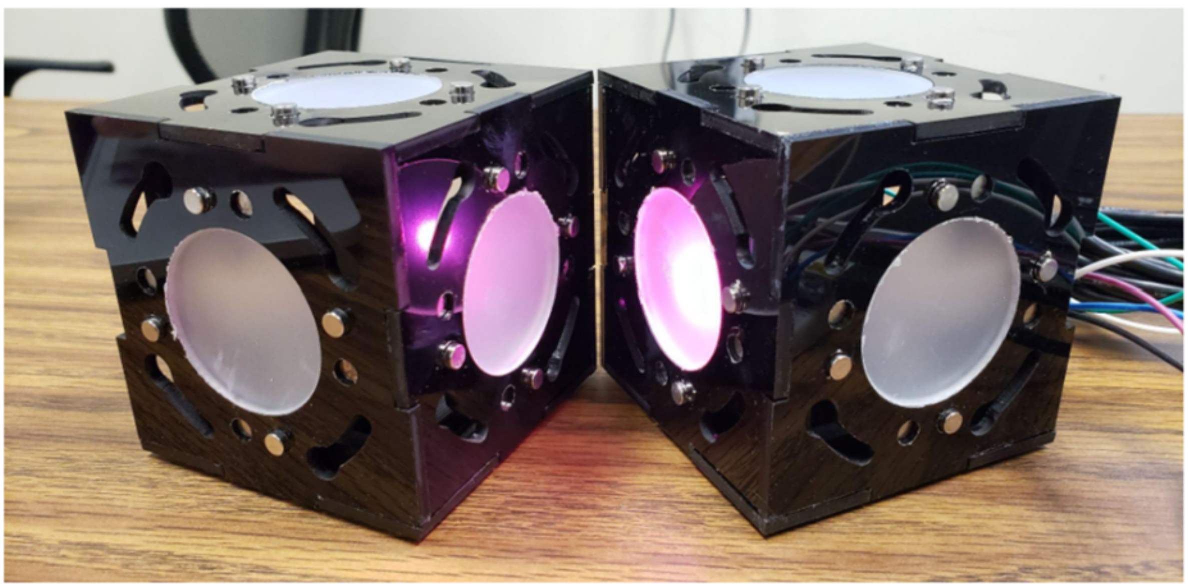
Figure 9. CAD model of the proposed block design with 5 female sides and a single male side seen on the bottom of the left image [9]

We took inspiration from the heterogeneous block design due to its greater shear load capabilities but opted for a stronger variant of the Allen key design with two different types of blocks depending on placement in the structure. The Allen key design was demonstrated by the SMAC 1.0 as being capable of creating large structures in simulation.

Finally, multiple potential methods of communication were discussed. We took particular interest in using near-field communication (NFC) protocol and IR LED communication. Although NFC was used in the original SMAC, we chose to use IR communication in our iteration. SMAC 1.0 laid much of the groundwork we build on today using their proof of concept as the foundation and benchmark for our new platform.

2.4.2 MARIA (SMAC 2.0)

Multi-Agent Robotic Intelligent Assembly (MARIA) shifted focus after the original SMAC project [7]. With more emphasis previously being put on the inchworm, MARIA opted to focus on the smart blocks instead. They used visual light communication to allow the blocks to understand where they exist in the space and to plan paths for construction. The blocks and the table of communication colors and their corresponding binary code are seen in Figure 10, with one of the lights used for communication turned on.



Binary Messaging Colors			
0000	0001	0010	0011
0100	0101	0110	0111
1000	1001	1010	1011
1100	1101	1110	1111

Figure 10. MARIA cubes and chart of the colors of their lights and corresponding binary code message. [7]

The types of connections between blocks were also experimented with to find the strongest that the inchworm could still easily navigate. Through experimentation with different load weights and configurations, it was determined that magnets were the best option. Some examples of possible configurations with the magnet connections can be seen in Figure 11. We continued using the magnet connections in our design as a form of weak connection between blocks. This still allows blocks to connect where the new heterogeneous design would not allow strong connections.

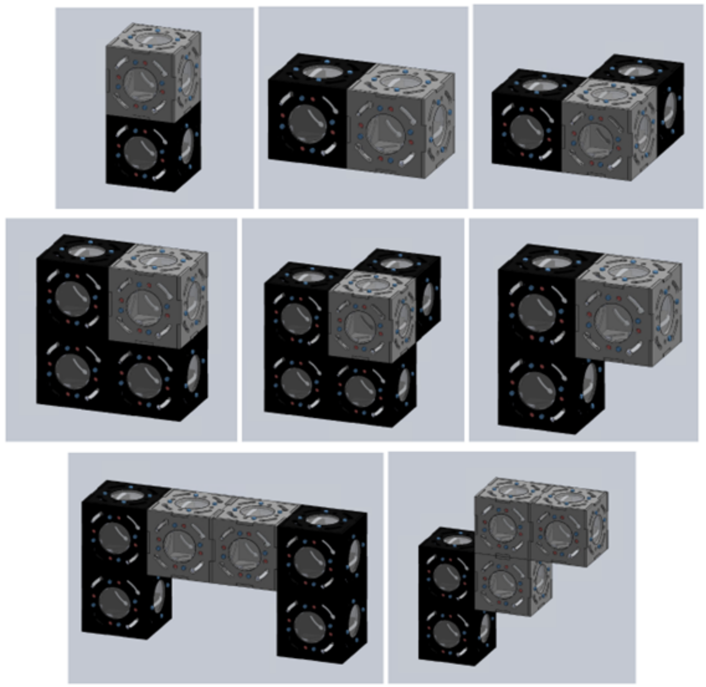


Figure 11. Some of the potential structures that are possible with the magnet connections

Although the general design of the MARIA blocks will be maintained in the new iteration, a new form of communication will be used. Instead of visual light communication, infrared (IR) communication will be used. It is cheaper to produce than using visual light, and it is also more reliable.

2.4.3 RAINSTORM

Rooftop Assembly Inchworm Network and Swarm Tiling Optimization Rooftop Maintenance (RAINSTORM) utilized the work of previous iterations of SMAC to autonomously install shingles on a rooftop [10]. Like MARIA, RAINSTORM aimed to implement a decentralized control algorithm by storing and communicating data through the shingles. The project introduced an inchworm inspired by the SMAC 1.0 inchworm, featuring a new end effector to manipulate the shingles. Electro-permanent magnets were used so that they could be enabled and disabled based on whether or not they were picking up or placing a shingle. The inchworm used in previous iterations was redesigned and fully rebuilt for use with the new smart shingles.

A series of images showing a demonstration of the inchworm putting a shingle can be seen in Figure 12.

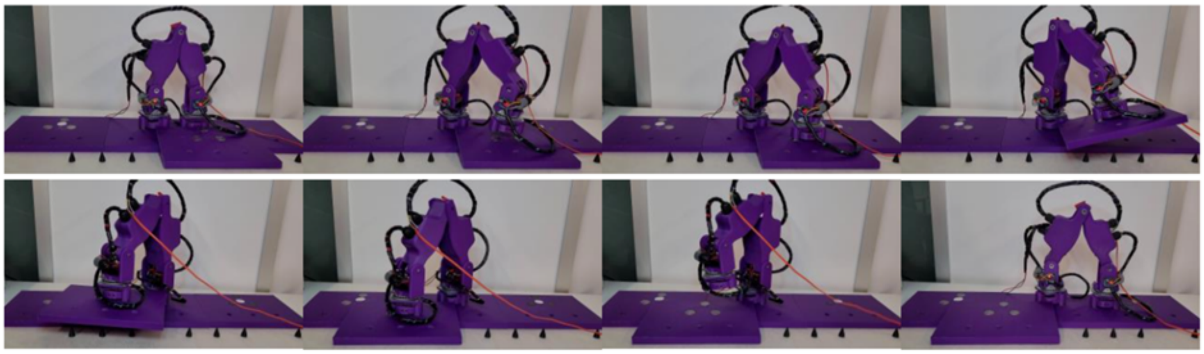


Figure 12. Demonstration of inchworm placing a shingle [10]

SMAC 3.0 takes a different direction from RAINSTORM, iterating on the smart block design developed by MARIA. However, we have chosen to continue using the inchworm developed by RAINSTORM based on the movement capabilities demonstrated on the shingling platform. We also found the range of motion well within the constraints needed for use with smart blocks. From the original inchworm design created by the SMAC 1.0 team, RAINSTORM altered the dimensions of each linkage as shown in Figure 13 to increase strength, support additional electronics, and prevent collisions between each link. To ensure the kinematics of the original design were preserved, the lengths of Links 1 and 4 were increased the same amount that Links 0 and 5 were decreased.

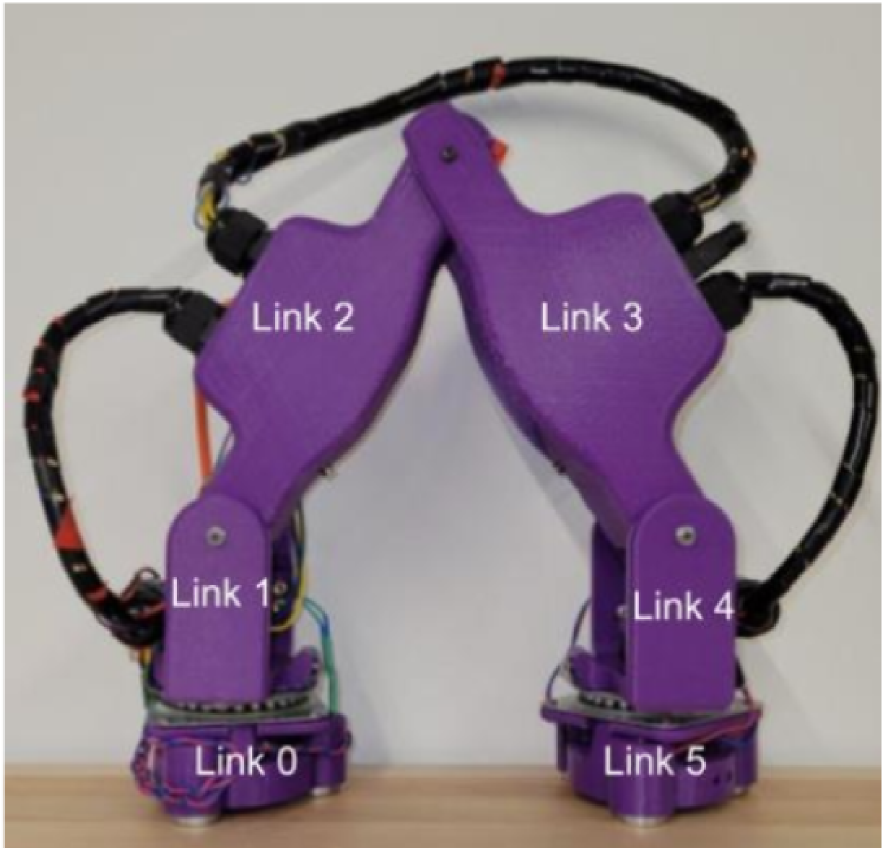


Figure 13. RAINSTORM altered linkage design with labeled links [10]

The ankles of the inchworm (links 1 and 4) were also redesigned to be stronger than the previous iteration, and the width of links 2 and 3 were increased due to pressure from the PLA casing causing faults from the encoders within. The design overhaul completed by RAINSTORM allowed us to easily translate the inchworm and end effector design back to smart block construction.

Chapter 3: Design and Development

Design Requirements

The SMAC 3.0 construction platform aims to reduce the gap between robotic swarm construction research and real-world application. SMAC 3.0 prioritizes strength and stability in the structures it builds to better represent realistic structure constraints. The construction system used an adapted version of the RAINSTORM inchworm robot as the manipulator agent and improved smart building block agents, all controlled by a novel algorithm.

The SMAC 3.0 team identified the following requirements for each subsystem of the platform:

- *Smart Block Mating:* the mechanical mate between interconnected building blocks should demonstrate higher strength, specifically shear loading resistance. As a baseline, MARIA's iteration of the smart blocks were able to withstand about seven pounds of force laterally and about two pounds of force in shear. Our improved linkage system should raise the shear strength of a connection by at least doubling the maximum loading capabilities of the last iteration of blocks used in the MARIA project.
- *Smart Block Structures:* The MARIA project found that their system could not create structures with overhangs of more than three blocks. Further, the weight limitations derived from the magnetic mating of the blocks made "inverted staircase" structures of more than three blocks in width impossible. In conjunction with an improved mating system, multi-block structures on the SMAC 3.0 platform should demonstrate improved stability and capacity for certain substructures that were not possible before, such as arches, bridges, and overhangs of more than 3 blocks.
- *Blueprint Generation and Structure Testing Algorithm*: the new control algorithm should maintain decentralized operation as was introduced by the previous iteration of SMAC and improve on stigmergic behavior with respect to the intelligent agents of the SMAC 3.0 system. In the scope of this project, to say the algorithm is complete is to say it can guarantee a physically stable solution for the input structure with the prescribed weight and stability constraints from our smart block hardware. The novel algorithm should account for strength and stability constraints that complement the mechanical improvements to the smart block. A successful implementation of new stability

algorithms and the improved smart blocks should demonstrate the ability to create structures that exceed past projects' limitations.

- *Communication System*: A reliable and improved communication protocol between the blocks should demonstrate reliability and low transmission time, which were limitations of the MARIA project and their visual light communication system. The MARIA system communicated at a rate of 16.5 bytes/s as a baseline. The new system should demonstrate comparable or faster speeds and robustness.

3.1 System Overview

Like its predecessors, SMAC 3.0 is designed with two collaborative agents of construction: the inchworm robot and the smart blocks. Our work emphasizes the development of the smart blocks and the blueprint construction algorithm. The inchworm robot for our system is from the RAINSTORM project with a new end effector to interface with the smart blocks. An image of the system is shown in Figure 14.

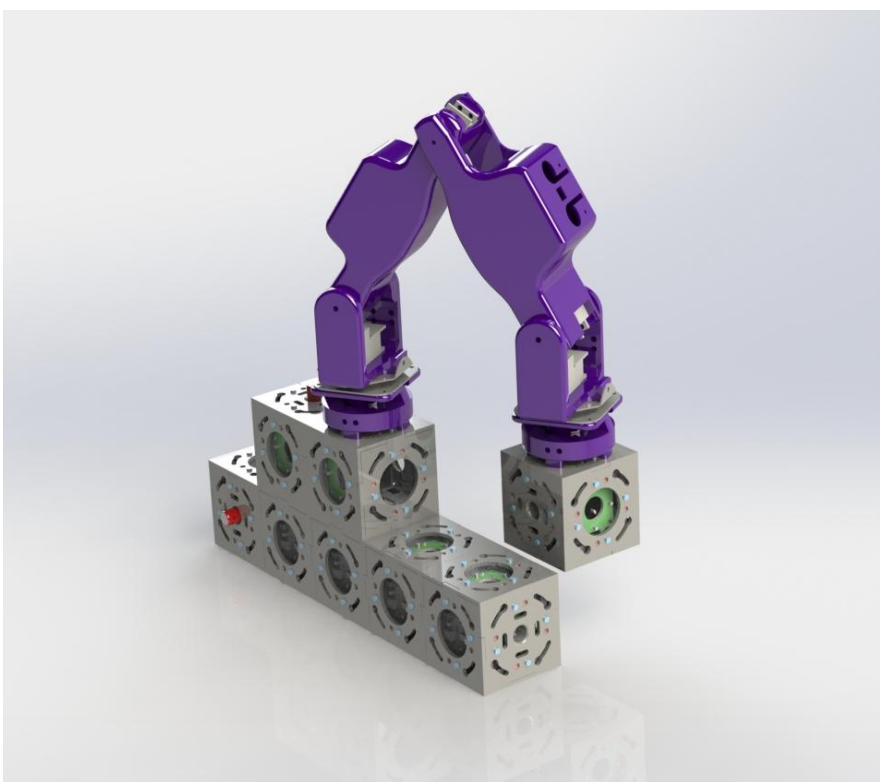


Figure 14. SMAC 3.0 system consisting of the new smart blocks and RAINSTORM inchworm.

The SMAC 3.0 smart blocks communicate through IR that allows for data to flow between blocks, as well as between a block and the robot. Our new smart blocks feature a backbone linkage system, which is created with a series of removable magnetic pegs that support higher shear loading between connected blocks. The backbone system creates two forms of mechanical mates between blocks, which we call “strong” links (presence of a backbone peg in the connection) and “weak” links (lack of a backbone peg in the connection). Further, we limit a single block to create up to two strong links with adjoining blocks, constrained at 90° and 180° orientations. The final smart block is shown in Figure 15.

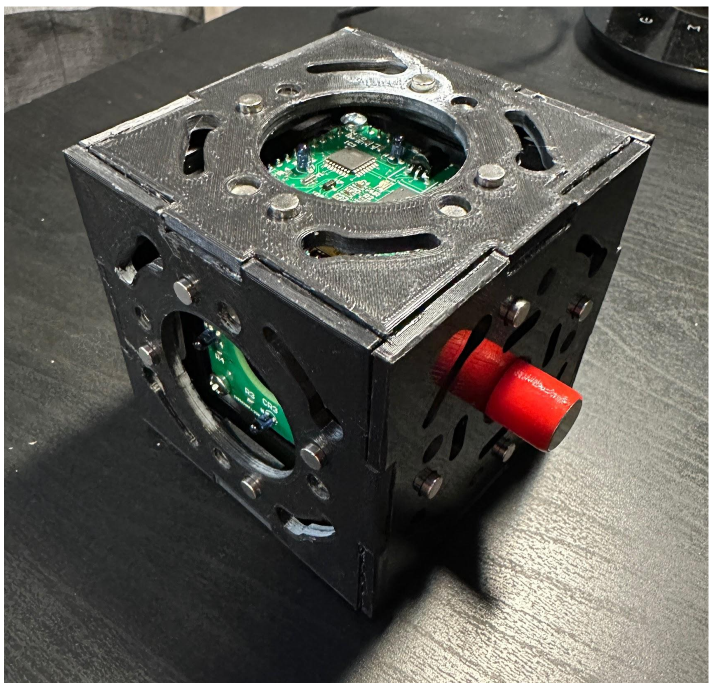


Figure 15. SMAC 3.0 smart block fully assembled, with a peg inserted.

The novel blueprint construction algorithm we developed works in conjunction with the new smart blocks, and validates a given input structure with real-world constraints for strength and stability. Our algorithm solves for strong link (backbone) placement within the structure, and after validation, outputs a “blueprint” for construction containing information regarding block position, orientation, and identifiers for neighboring blocks. The algorithm guarantees that a given structure is possible to build (in real life) after passing validation. Should the structure violate a constraint, the algorithm identifies where in the structure and which block is responsible for the point of failure. Structures built using our platform demonstrate increased strength and stability over past iterations of the project.

3.2 Block Improvement Approach

Inspired by the smart blocks developed in the MARIA project, SMAC 3.0 focuses on two main areas for improvement regarding the smart blocks:

- *Shear Loading Resistance:* The new blocks should incorporate a new mechanism to support shear loading between blocks, which allow the structures created to be more stable during and after construction.
- *Potential Structure Complexity:* The new block design should allow for stronger connections when overhanging a set amount of blocks, necessary to build common architectural features like doorways and ceilings.

To properly fulfill the design requirements established for the smart blocks, SMAC 3.0 implements an improved magnetic linkage system that allows for any block to create "strong" links and "weak" links between block faces as needed. This system will enable higher structural stability and also integrate with the smart blocks and be able to withstand higher loading forces while not compromising the modularity of the system. Using this concept, SMAC 3.0 created three block prototypes that fulfilled the design requirements. As with any design process, each of these prototypes had advantages and disadvantages and must be carefully assessed to find which block design had the best balance of compromise.

3.2.1 Permanent Peg Blocks

The first block design featured a magnetic extruding peg that would significantly raise the shear strength of all strong-linked connections, shown in Figure 16. These blocks offer strong-link connections on two faces while allowing standard-links on all other faces of each block. This design features two configurations for the strong-link faces based on the orientation of the strong linking: 90° and 180°. This allows the formation of strong-link chains that can perform internal 90° turns according to the structure's stability requirements. The main issue with this design is that the placement of blocks might be difficult for the robot due to the permanently protruding pegs being irremovable by the inchworm from certain angles, and orientation constraints would slightly complicate the building algorithm.

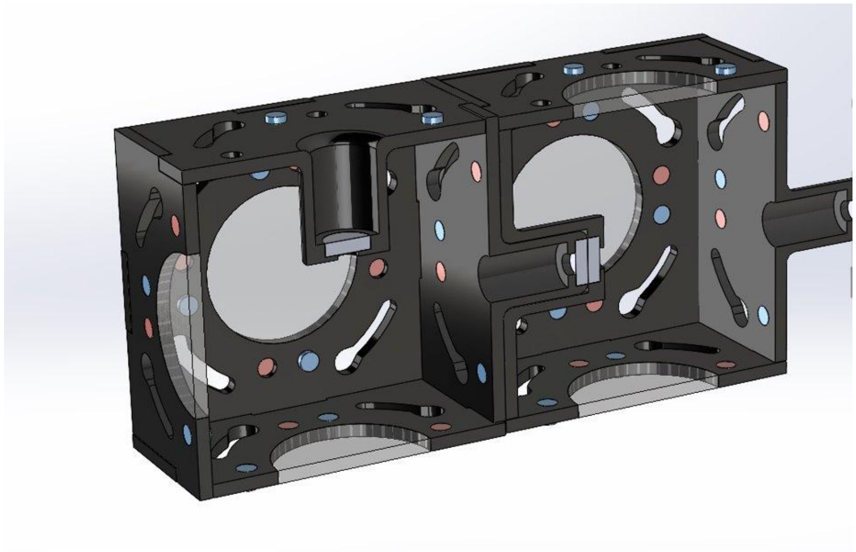


Figure 16. Permanent Peg Design; 90° block (left) and 180° block (right), section view.

3.2.2 Removable Peg Blocks

This design offers five strong-linkable faces and one standard-linkable face, as shown in Figure 17. These blocks can generate strong, shear-resistant mates through a removable magnetic peg that can withstand higher shear loads and create webs of strong-links to support even more complex structures in the areas that require the most support. In this design, the pegs are removable, which will make it easier for the robot to place, and also keeps the blocks more streamlined. The removable pegs are separate parts from the blocks and need to be placed. For the scope of our project, the team decided not to solve the problem of peg placement, and instead for our final demonstration, the pegs will be inserted manually through human interaction. Also, the orientation constraints of these blocks will complicate our building algorithm slightly more than the Permanent Peg Design.

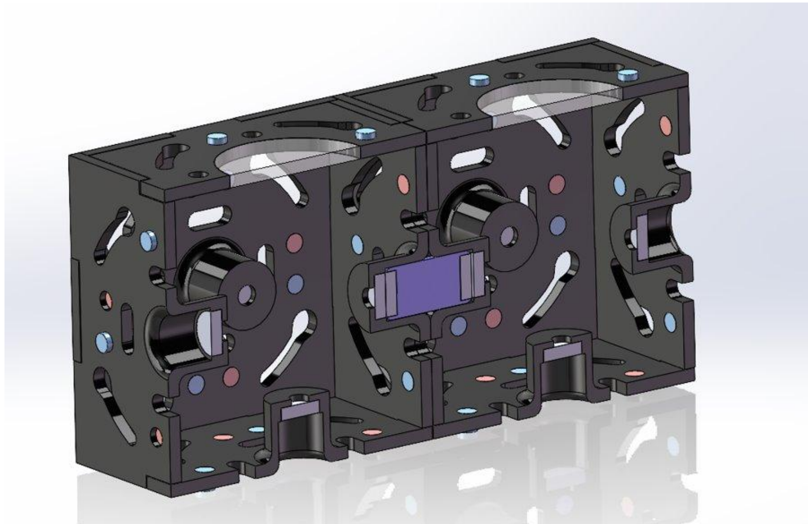


Figure 17. Removable Peg Design, section view.

3.2.3 Hybrid Blocks

By applying the strengths of the first two block designs, the team created the Hybrid block design shown in Figure 18. The hybrid design features a similar strong-link mating system constrained at 90° and 180° orientations from the Permanent Peg design, along with the removability of the pegs from the Removable Peg design, to create an efficient block with balanced compromises. The trade-off with this design is the same as the Removable Peg design: the pegs need to be placed and for the scope of this project, will need to be inserted manually. Additionally, the orientation constraints will complicate the algorithm slightly, just like the permanent peg block configuration.

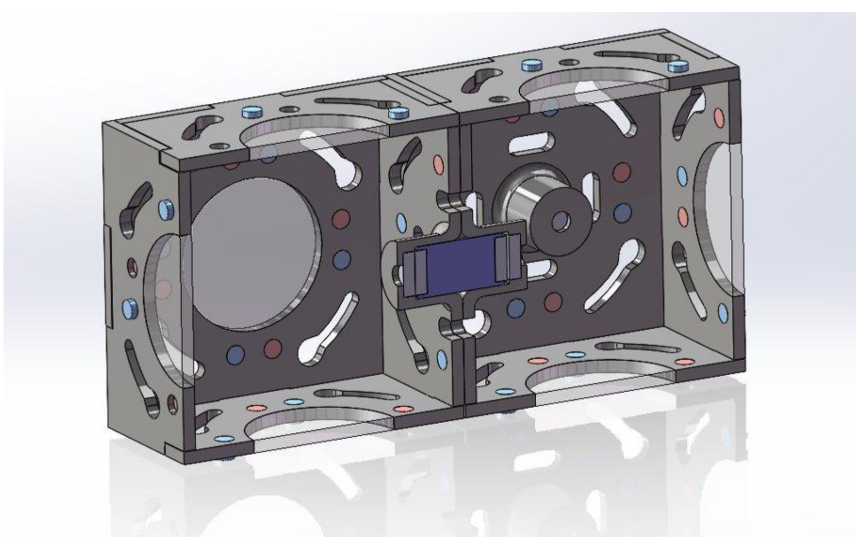


Figure 18. Hybrid Block Design, section view.

3.3 Block Design Testing

With three block designs, thorough testing of each design is essential in making an informed decision for the SMAC 3.0 smart blocks. Mechanical testing was split into two design groups: permanent pegs and removable pegs (not to be confused with the design names). Since the Hybrid Block design shares the same peg dimensions as the Removable Pegs design, the mechanical testing for these two designs were placed under the same group. The mechanical tests involved force testing to measure the strength of the mating system in two loading conditions, as well as testing to verify the compatibility and strength of the blocks arranged in ways that reflect real-life components of a structure.

3.3.1 Force Testing

To empirically test the strength of the mating system designs, blocks were tested for lateral removal force (in the direction of the mate) and shear loading force (perpendicular to the direction of the mate). As a baseline, MARIA's iteration of the smart blocks were measured to withstand an average of 6.7 pounds of force laterally, and an average of 1.7 pounds of force in shear.

Two separate sets of trials were performed for each of the design groups. The latest trial set is shown in Table 1. The forces were measured using a small force scale and attached with a hook to a string tied to the blocks such that the force was evenly centered on the face.

Table 1. Force Testing for Permanent Pegs and Removable Pegs Design Groups

Test Type	**Test 1**	**Test 2**	**Test 3**	**Test 4**	**Test 5**	**Average**
Perm. Pegs						
Force, lateral (lbf)	7.89	7.36	7.4	7.98	8.13	7.75
Force, shear (lbf)	-	-	-	-	-	10+*
Rem. Pegs						
Force, lateral (lbf)	10.29	10.89	11.65	8.49	10.2	10.30
Force, shear (lbf)	7.88	7.85	6.88	7.73	6.54	7.38**

* Can consistently support 10+ lbs of shear loading. Face magnets slightly separate at ~3 lbs. Not tested for absolute failure.

** Force reflects a distinct disconnection of face magnets but the peg remained secure. Not tested for absolute failure.

From the testing, the team found that both design groups outperformed the previous version of the smart blocks from the MARIA project by a substantial margin. The main takeaway from these tests was the ability of these block designs to withstand much higher shear loading, which was a limitation on the standard MARIA blocks. A higher capability for this loading condition is vital to having stable and strong structures.

3.3.2 Structure Testing

For the structure testing experiments, we decided to perform force tests on various substructures using multiple blocks for each design group. The MARIA project identified eight substructures that are critical to constructing real structures. These include substructures like overhangs, corners, or doorways. The eight substructures that this project tested are shown in Figure 19. The backbone/strong-link configurations tested for each design group are shown in red in the section views of the blocks.

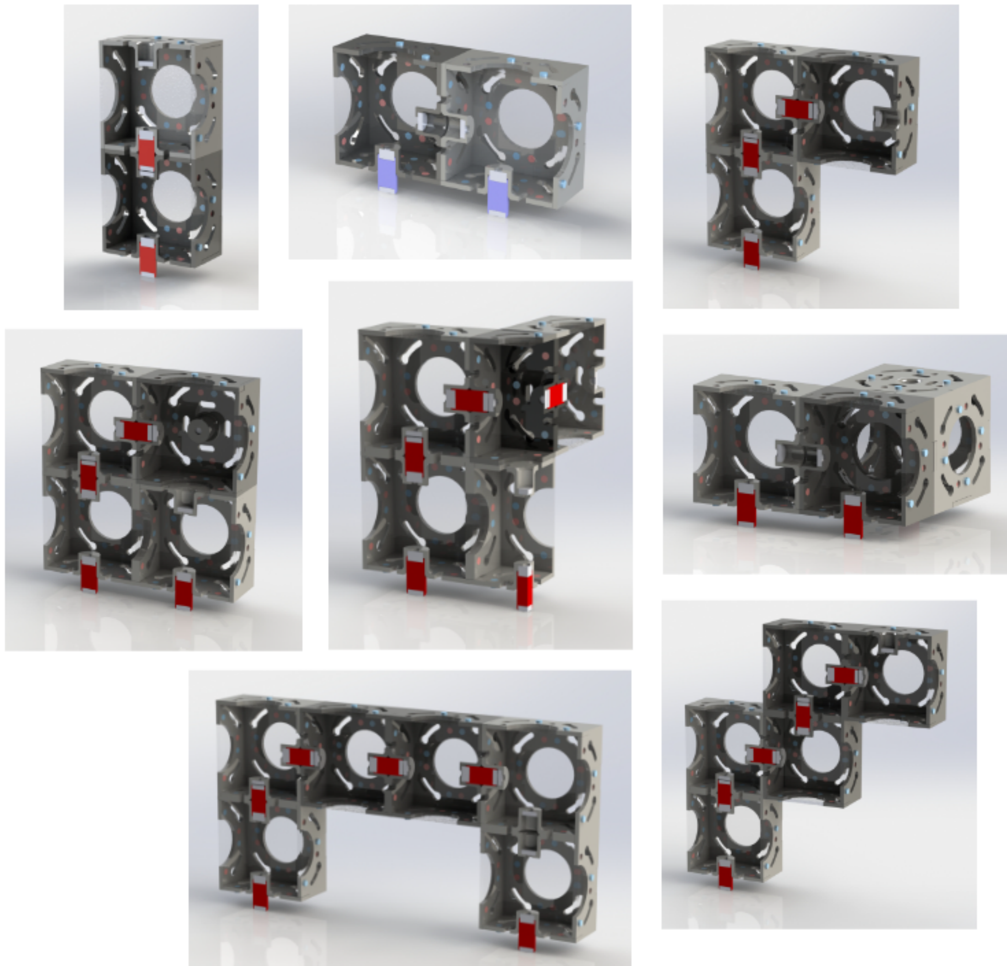


Figure 19. Basic substructures with strong-link configurations colored in red. From left to right: Simple Top, Simple Side, Vertical Corner, Vertical Corner 2, Vertical Corner 3, Ground Corner, Overhang, Doorway, Inverted Staircase.

3.3.2.1 Structure Compatibility Testing

The first step in structure testing was to test the compatibility of the block designs with the various substructures. Through compatibility testing, insights on the effect of tolerance stack-up (i.e., the cumulative error of manufacturing inconsistencies) between several blocks and block placement with protruding features were gained. The data and observations for each design group are shown in Table 2.

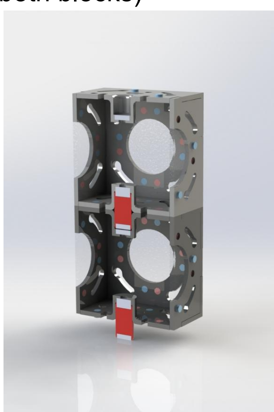
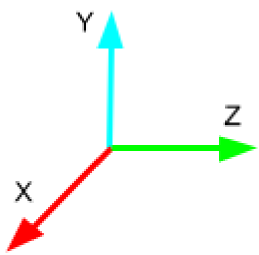
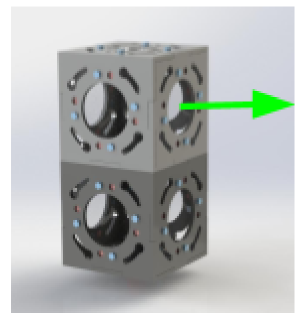
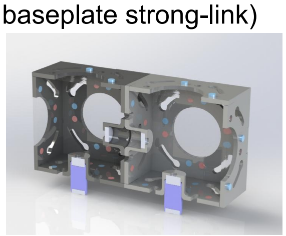
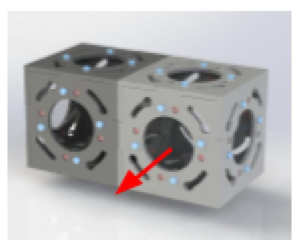
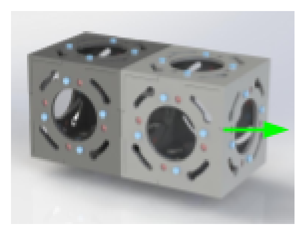
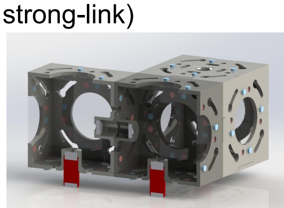
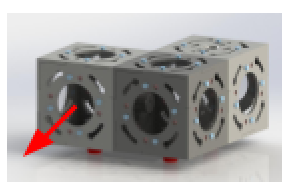
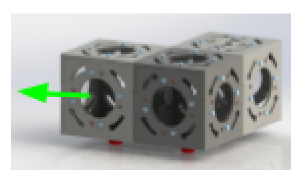
Table 2. Block Design Structure Compatibility

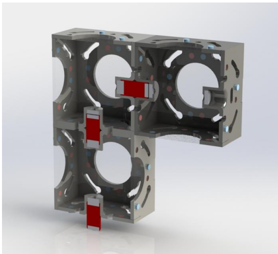
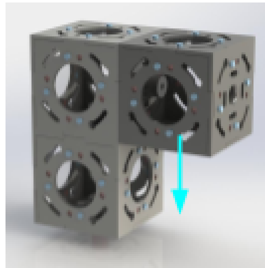
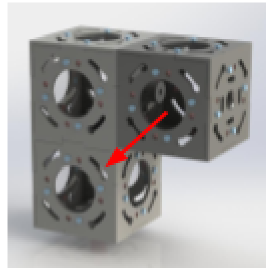
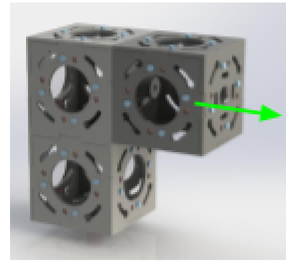
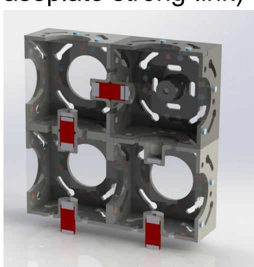
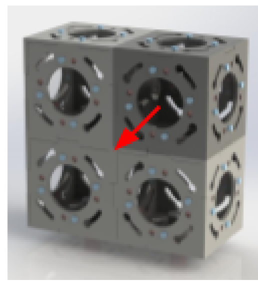
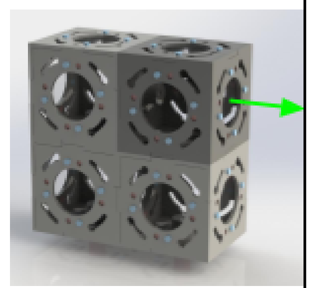
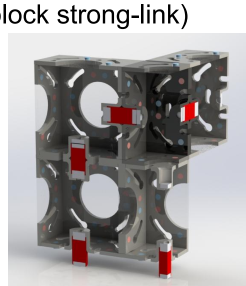
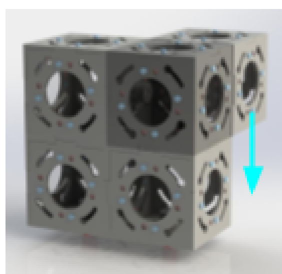
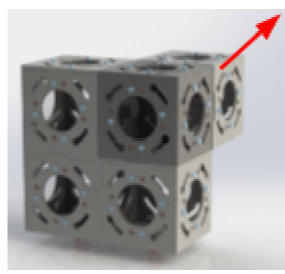
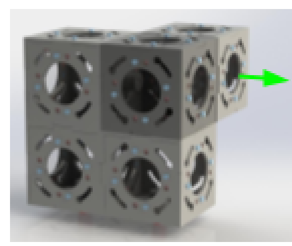
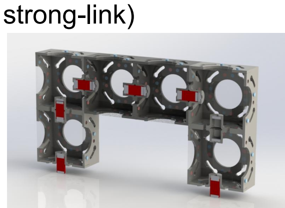
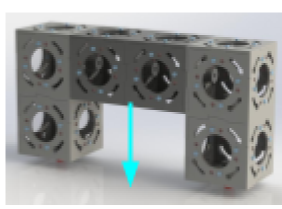
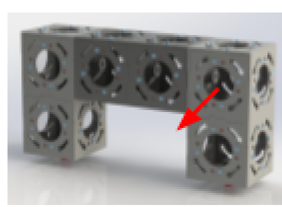
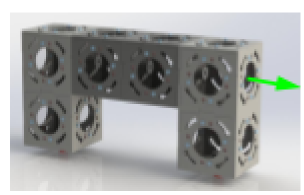
	Design Group			
Substructure	Permanent Pegs	Notes Perm. Pegs	Removable Pegs	Notes Rem. Pegs
Simple Top	TRUE		TRUE	
Simple Side	TRUE		TRUE	
Ground Corner	TRUE	Corner cannot be placed if the other two go down first and strong links are horizontal; fine if strong links are into ground	TRUE	
Vertical Corner 2	TRUE	Needed minor wiggle	TRUE	Needed minor wiggle
Vertical Corner 3	TRUE	Needed minor wiggle	TRUE	Needed minor wiggle
Overhang (1 block)	TRUE		TRUE	
Overhang (3 blocks)	TRUE		TRUE	
Max Overhang #	6	Stability @ base block decreases	5	Stability @ base block decreases
Doorway	TRUE	Needed wiggle/finesse to place last corner	TRUE	Needed wiggle/finesse to place last corner
Extended Doorway	TRUE		TRUE	
Inv. Staircase (3W)	TRUE		TRUE	
Inv. Staircase (4W)	TRUE	Stability @ base block decreases	TRUE	Stability @ base block decreases

3.3.2.2 Structure Load Testing

The next stage of structure testing was static structure load testing, which gauges the ability of an assembly of blocks to resist structural failure under static load. Using a force probe, the team was able to identify the strength required to reach failure in up to three loading conditions depending on the substructure and the potential stresses that a substructure would need to tolerate during realistic loading. Realistic loading includes the static load from the weight of the structure itself as well as the weight and dynamic load from the inchworm construction agents. First, the team identified the critical points within each substructure and chose to apply loads to the block that would affect that critical connection the most. Then, the same substructures and critical connections were tested for each design group using the same loading conditions. Each test consisted of five trials. Finally, we analyzed our results to determine what were the advantages and disadvantages of each design group. The data for the permanent peg design group is shown in Table 3.

Table 3. Substructure Static Load Data for Permanent Pegs Design Group

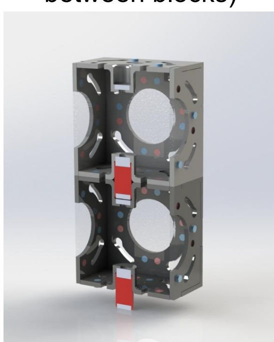
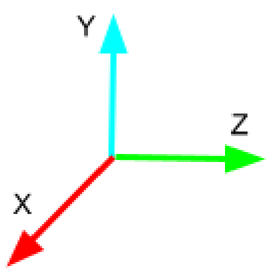
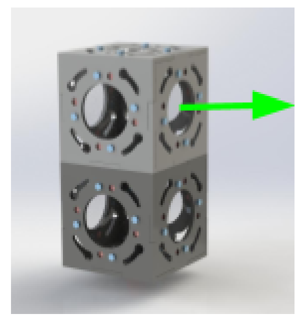
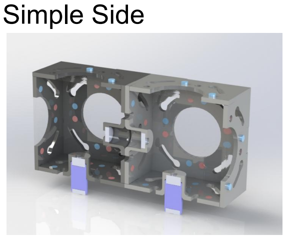
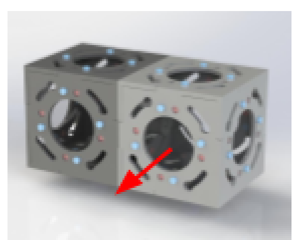
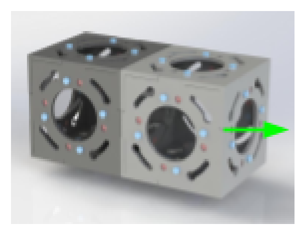
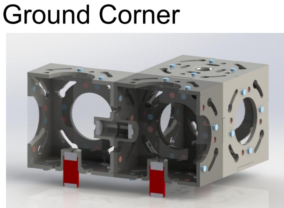
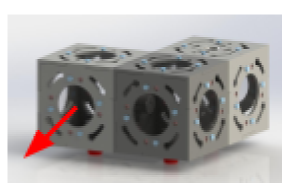
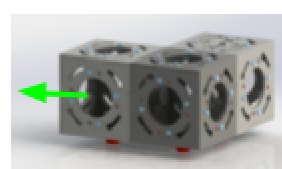
Substructure: Using Permanent Pegs	**Y-Direction Shear Loading**	**X-Direction Shear Loading**	**Z-Direction Tensile/Shear Loading**
Simple Top: (Failure at the strong-link between both blocks) 	**Coordinate Axis** 	-	 Average: 3.67 lbs
Simple Side: (Failure at baseplate strong-link) 	-	 Average: 25+ lbs	 Average: 12.14 lbs
Ground Corner: (Failure at baseplate strong-link) 	-	 Average: 25+ lbs	 Average: 12.14 lbs

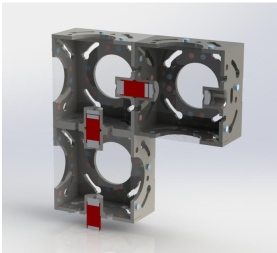
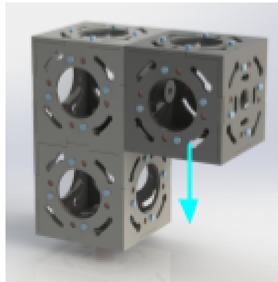
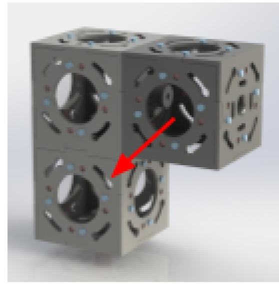
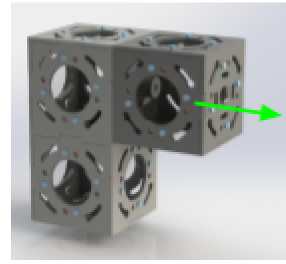
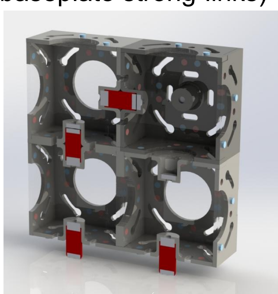
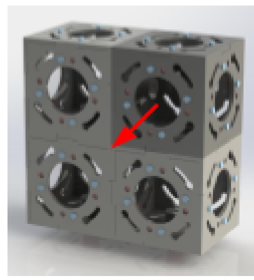
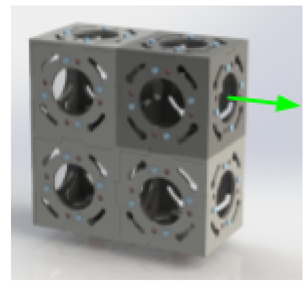
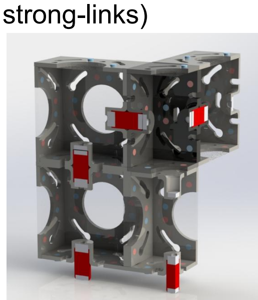
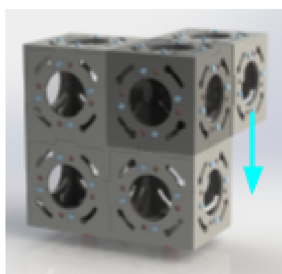
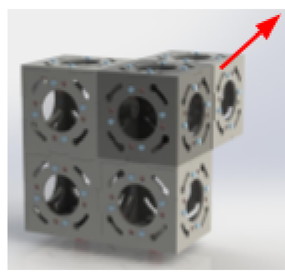
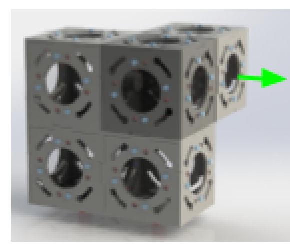
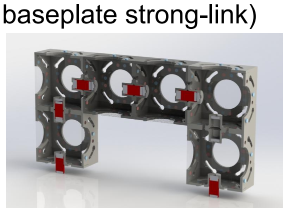
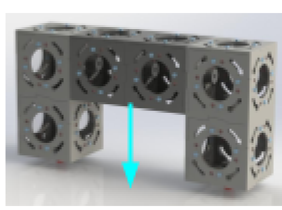
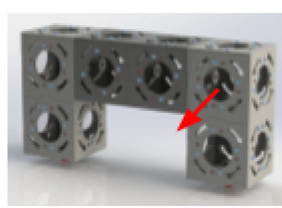
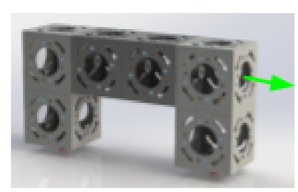
Vertical Corner: (Failure at baseplate strong-link) 	 Average: 10.00 lbs	 Average: 3.23 lbs	 Average: 7.10 lbs
Box: (Failure at baseplate strong-link) 	-	 Average: 8.90 lbs	 Average: 16.21 lbs
Box with Corner: (Failure at overhang block strong-link) 	 Average: 10.04 lbs	 Average: 7.09 lbs	 Average: 3.21 lbs
Doorway: (Failure at baseplate strong-link) 	 Average: 25+ lbs	 Average: 11.12 lbs	 Average: 15.42 lbs

From the structure load testing on the permanent pegs design group, the data in Table 3 shows that the design functions as intended and indeed increases strength in multi-block substructures. As these substructures are common in larger, more complex structures, the performance data also serves as proof of concept that the permanent peg design is capable of stronger, more stable structures than what was previously possible.

We performed the same procedures for static load testing on the substructures constructed with the permanent pegs design group to compare both of our new design groups. The test data is shown in Table 4. Similar to the permanent pegs design group, the removable pegs design class performed well, indicating that these smart blocks are capable of constructing stronger and more stable structures in comparison to the substructures that the MARIA blocks were able to achieve.

Table 4. Substructure Static Load Data for Removable Pegs Design Group

Substructure: Using Removable Pegs	**Y-Direction Shear Loading**	**X-Direction Shear Loading**	**Z-Direction Tensile/Shear Loading**
Simple Top (Failure at strong-link between blocks) 	**Coordinate Axis** 	-	 Average: 3.57lbs
Simple Side 	-	 Average: 25+lbs	 Average: 25+lbs
Ground Corner 	-	 Average: 25+lbs	 Average: 25+lbs

Vertical Corner (Failure at baseplate strong-link) 	 Average: 7.98lbs	 Average: 3.36lbs	 Average: 3.88lbs
Box (Failure at both baseplate strong-links) 	-	 Average: 6.86lbs	 Average: 15+lbs
Box with Corner (Failure at overhang Block strong-link and both baseplate strong-links) 	 Average: 7.39lbs (Failed at overhang block strong-link)	 Average: 6.80lbs (Failed at baseplate strong-links)	 Average: 6.96lbs (Failed at overhang block strong-link)
Doorway: (Failure at baseplate strong-link) 	 Average: 25+lbs	 Average: 5.44lbs	 Average: 8.51lbs

3.3.3 Design Decision

After thorough structure testing of the different design groups, the team decided to implement the Hybrid block design, seen in Figure 20, because it has the best balance of compromise out of all three of our new designs. At the beginning of the block design process, the current project team focused on improving the shear load capacity and increasing the number of complex structures the blocks could form. In testing, both design groups passed compatibility with the designated substructures, which ensured that the structures could be built by the inchworm agent after implementation. Next, the team tested various substructure configurations until failure in a static load test. Both design classes yielded results that indicate improved strength in strong-linked connections and overall structural stability. An important point to consider when analyzing the block designs' performance is the inchworm agent's capability to create the structures in the real world. Thus, qualities like easier robot manipulation (from non-protruding and removable pegs) or strong-link orientation constraints in the final block design are important considerations as opposed to blindly choosing designs based on performance scores. That said, the advantages of the Hybrid Block design and its strong test performance make it an ideal choice to implement in the final SMAC 3.0 construction system.

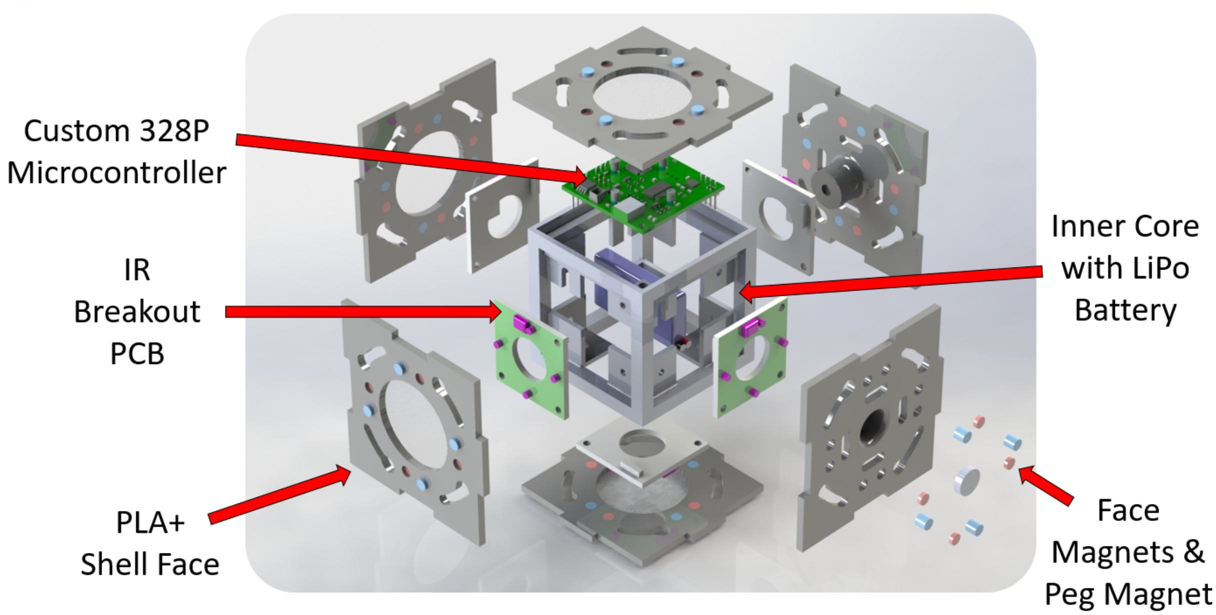


Figure 20. Exploded assembly of final block design and its components.

3.4 Communication System & Electronics

The communication system must establish reliable communication between the blocks for block-to-block and block-to-robot communication. It should demonstrate scalability, low error, and low transmission time. Past iterations of the SMAC platform used NFC or VLC for communication between blocks. The communication method of choice for SMAC 3.0 is through infrared (IR) communication, using the 38kHz NEC-standard transmission protocol. IR communication is simple, inexpensive, and is proven to be reliable in numerous use cases, such as a standard television remote. The NEC infrared transmission protocol is depicted in Figure 21 [11]. This protocol, when used as intended with the address space for device identification, stores 1 byte of data within the command space per message.

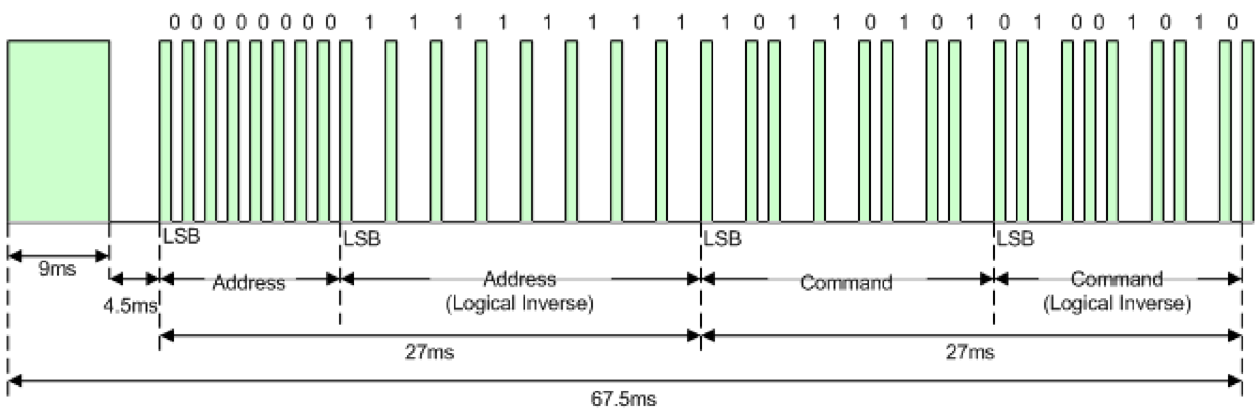


Figure 21. Message frame for the NEC protocol, courtesy of Altium Ltd. [11]

The IR communication method only requires two components: an IR-emitting LED, and an IR-receiving sensor. The IR sensor of choice is a VISHAY TSOP38438 sensor, which has internal components that demodulate the 38kHz carrier pulse, such that the output signal can directly connect to a microcontroller for use. Further, this sensor offers resistance to ambient light sources - improving robustness and reliability of the system. To accommodate the central peg for the blocks’ mechanical mating system, the breakout boards that host the IR communication components have through-holes. Each block face has a breakout board for communication, totaling five breakout (or “child”) boards with one main board with the same communication components that also host the microcontroller for the block. The integration of the breakout boards with the blocks is shown in the CAD rendering in Figure 22.

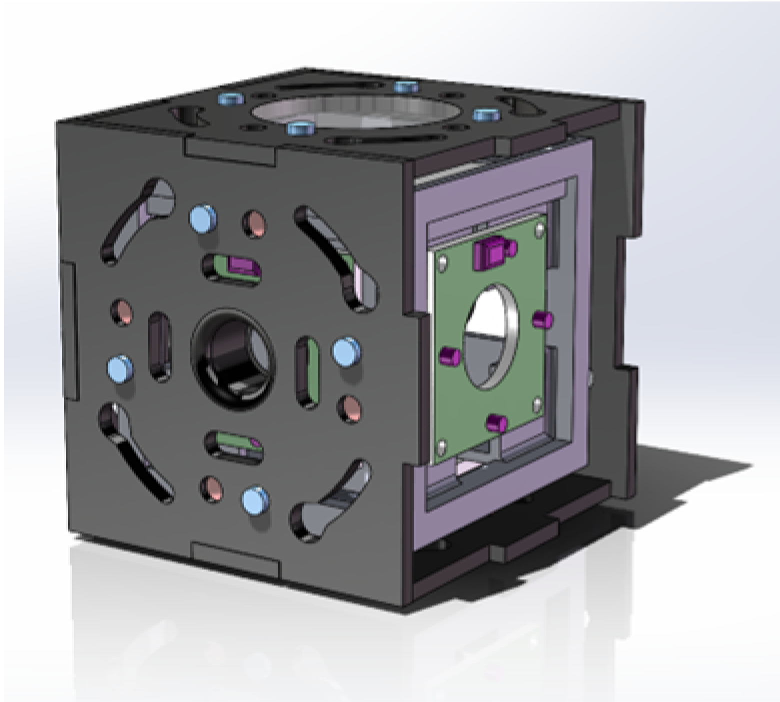


Figure 22. Hybrid block design with breakout board mockups.

3.4.1 IR Communication Testing

To verify the compatibility of the selected IR communication components with our project, a series of simple communication tests were performed with two pairs of transmitters and receivers. These tests consisted of a simple handshake test, handshakes at varied distances, and handshakes with unfrosted, frosted, and two frosted acrylic discs in between. The testing setup is shown in Appendix B. handshake was deemed successful when the received message matched the transmitted message. The testing was successful, and all tests passed with no issues. The data for these tests are shown in Table 5.

Table 5. Communication Handshake Tests

Handshake Test		**Separation Test**			**Acrylic Windows**					
					Frosted		Unfrosted		Double Frosted	
Trial	**Success**	**Trial**	**Distance (cm)**	**Success**	**Trial**	**Success**	**Trial**	**Success**	**Trial**	**Success**
1	TRUE	1	1	TRUE	1	TRUE	1	TRUE	1	TRUE
2	TRUE	2	5	TRUE	2	TRUE	2	TRUE	2	TRUE

3	TRUE	3	10	TRUE	3	TRUE	3	TRUE	3	TRUE
4	TRUE	4	15	TRUE	4	TRUE	4	TRUE	4	TRUE
5	TRUE	5	20	TRUE	5	TRUE	5	TRUE	5	TRUE
6	TRUE	6	25	TRUE	6	TRUE	6	TRUE	6	TRUE
7	TRUE	7	30	TRUE	7	TRUE	7	TRUE	7	TRUE
8	TRUE	8	35	TRUE	8	TRUE	8	TRUE	8	TRUE
9	TRUE	9	40	TRUE	9	TRUE	9	TRUE	9	TRUE
10	TRUE	10	45	TRUE	10	TRUE	10	TRUE	10	TRUE

Further testing for the IR communication components was conducted for sequences of 4-byte encoded NEC messages that better reflect real data transmission. Sequence testing consisted of trials of 25 messages transmitting as fast as possible. The team found that the peak data transmission rate for the IR communication system was 14.8 byte/s for standard NEC messages, since the command space is 1 byte, and assuming the address space is strictly for device identification. The speed of transmission is largely limited by the ~70 ms it takes to send one encoded message - an intrinsic limitation of the NEC protocol. That said, creative encoding of data across both the address space and the command space will allow for additional data to be transmitted in one message, meaning a higher effective data transmission rate. As a benchmark, if each message contained 1.5 bytes of data (using half of the address space plus command space), the communication system can peak at 22.2 bytes/s. The speed and reliability from the IR communication testing proved to be more than satisfactory for the purposes of SMAC 3.0, and these tests served as a proof-of-concept.

3.4.2 Electronics

The smart blocks' electronics consist of a few major components: the microcontroller board, five IR breakout boards, and a lithium-polymer (LiPo) battery. The electronics for the smart block are secured to an inner core that is placed within the shells of each block. An assembled inner core is pictured in Figure 23. These components enable communication for all six faces of the smart block.

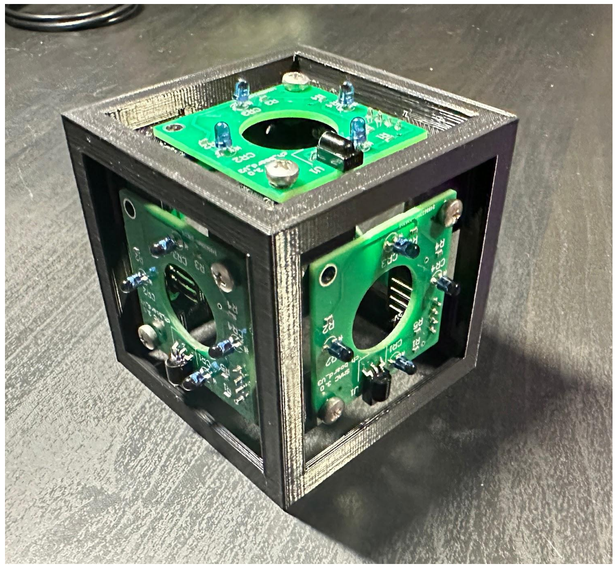


Figure 23. Smart block inner core with breakout boards attached.

The IR breakout board for each block face is shown in Figure 23. Four IR LEDs are arranged in a circle and one IR receiver is adjacent to one of the LEDs. This design allows for orientation independence in the communication system, such that no matter what orientation two blocks may be mated together, the IR receiver will always be aligned with an IR LED. These four LEDs are wired in parallel such that they all emit the same signal simultaneously without additional components.

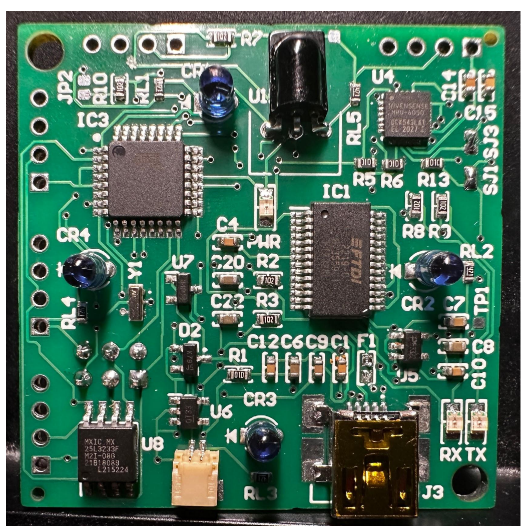


Figure 24. Custom 328p microcontroller board.

For the microcontroller board, the same IR components are included along with the microcontroller and other peripherals. Similar to the previous iteration of the block in the MARIA project, the ATMega328P microcontroller was chosen for its low cost and appropriate performance. The microcontroller board is a custom-layout design inspired by the Arduino Nano, integrated with the IR components, 32Mb flash memory, and a MPU6050 inertial measurement unit (IMU). The microcontroller board is shown in Figure 24. The breakout boards interface to the microcontroller board with only connections for ground, power, and two digital I/O signals. The electronics of the block are powered by a single-cell 600mAh LiPo battery and plugs into the beige JST-SH connector on the microcontroller board. A block diagram depicting the main peripherals of a smart block's electronics is shown in Figure 25.

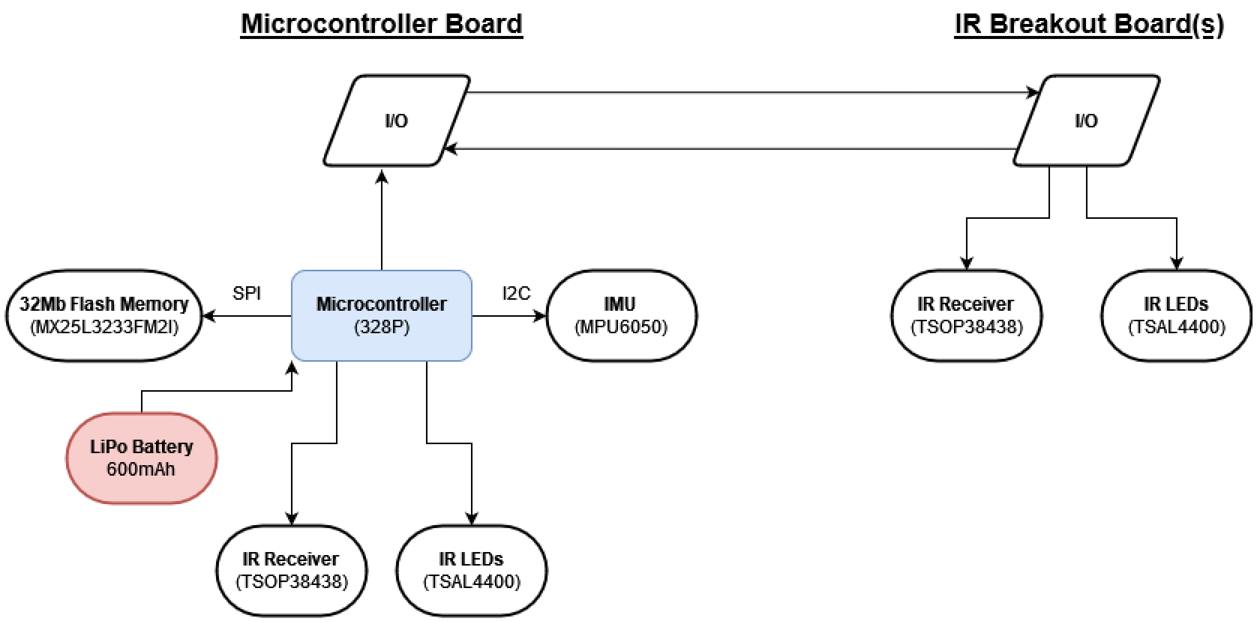


Figure 25. Block diagram of a smart block's electronics.

3.5 Blueprint Generation Algorithm

To demonstrate the capabilities of the inchworm construction and block system, we developed an algorithm to allow simulated construction of structures the new block and inchworm platform can build. By first simulated potential structures, we test whether a structure will fail in real world testing.

3.5.1 Design Requirements

The algorithm must be able to take any goal structure as input and validate the structure for weight and stability errors, then generate a blueprint - working in tandem with the new block design. In general, an algorithm is considered “complete” when it can guarantee a solution in a finite time if at least one solution truly exists [12]. For the purposes of SMAC 3.0, we consider an algorithm complete when it can guarantee a physically-stable solution for the input structure with the prescribed weight and stability constraints from our smart block hardware. Considering this distinction in the “completeness” of the algorithm, there may be conditions at which a solution to build the structure truly exists; however, the solution violates our imposed constraints and fails validation. Therefore, the design requirement for the SMAC 3.0 construction algorithm is that it is complete under our adapted definition of completeness, and not necessarily canonical algorithmic completeness.

3.5.2 Goal Structure Creation

To aid in the demonstration and flexibility of the algorithm, a simulation is used to allow any user to create a structure they would like to see built. This simulation was created in Python with the high-level design library Ursina [13] and customized by the MQP team. In Figure 26, the simulation is displayed with an example structure. This virtual environment guarantees no blocks can be placed out of bounds and no blocks can overlap the same position. The x, y, and z coordinates of each block are recorded upon exit of the simulation and sent directly to the algorithm for processing.

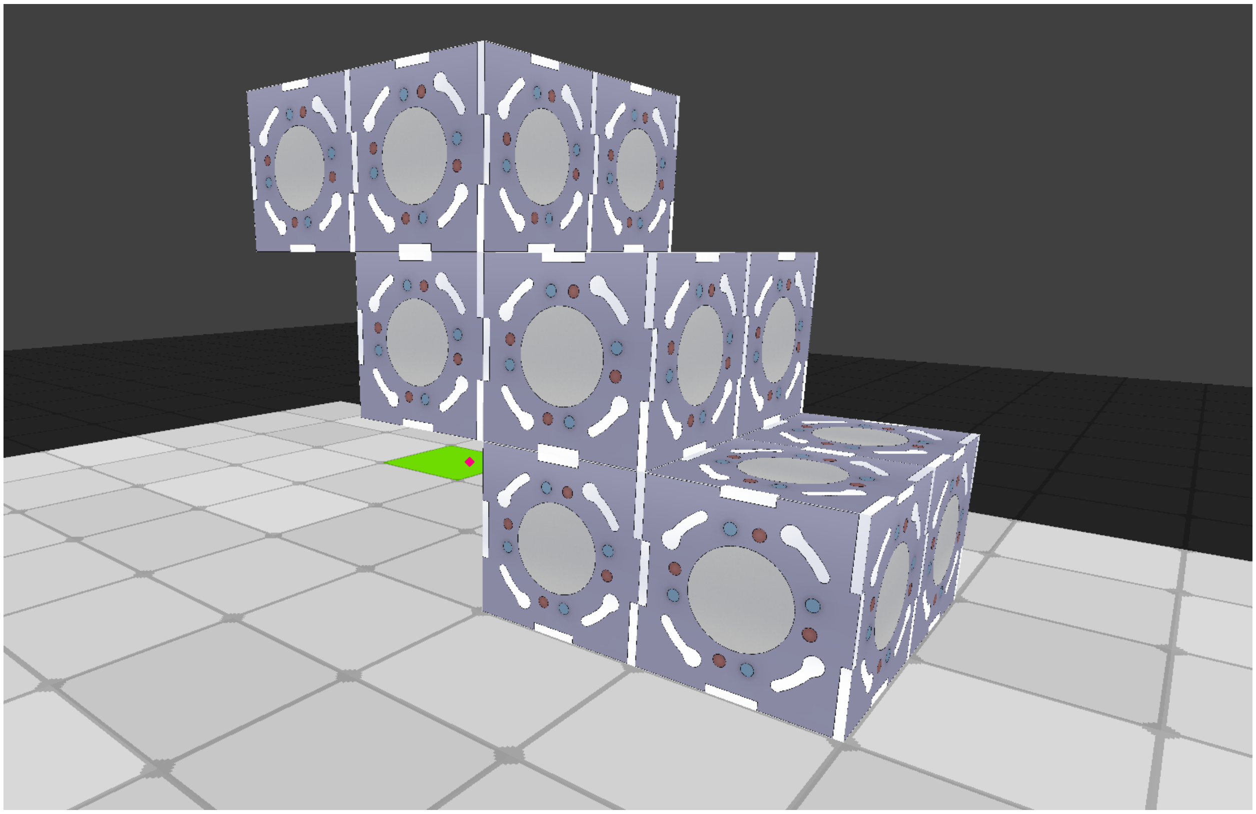


Figure 26. Construction Blueprint Simulation

3.5.3 Construction Metadata

Before the block communication blueprint can be generated, the order and orientation of all blocks must be assigned. For each block, metadata is generated for: their dependent block, the step of construction, the orientation of the strong joint (if applicable), and finally the type of block (90° or 180° block).

Generating the dependent block and step of construction are necessary to generate each block's orientation and test stability during the construction process. This order ensures that the blueprint does not try to place a floating block and that the placement of the block is reachable by an inchworm. During this process, the algorithm will notify the user and exit if it cannot properly assign all blocks a dependent and construction step. In Figure 26, the blocks on the floor do not have dependents, but all blocks above y=0 (with y as the vertical axis) must have a dependent so they are supported. The metadata is generated using a heavily customized breadth-first search algorithm on an edge graph of all blocks, with each vertex being a block with edges as their neighbors. In Figure 27, the output of the structure built in Figure 26 is shown.

```
ID:  0 XYS: [ 7 , 0 , 7 ] Crit: None  Depth: 0
ID:  1 XYS: [ 7 , 0 , 6 ] Crit: None  Depth: 0
ID:  2 XYS: [ 6 , 0 , 7 ] Crit: None  Depth: 0
ID:  3 XYS: [ 6 , 0 , 6 ] Crit: None  Depth: 0
ID:  4 XYS: [ 6 , 1 , 7 ] Crit: 2  Depth: 1
ID:  5 XYS: [ 6 , 1 , 6 ] Crit: 3  Depth: 1
ID:  6 XYS: [ 5 , 1 , 7 ] Crit: 4  Depth: 2
ID:  7 XYS: [ 5 , 1 , 6 ] Crit: 5  Depth: 2
ID:  8 XYS: [ 5 , 2 , 7 ] Crit: 6  Depth: 3
ID:  9 XYS: [ 5 , 2 , 6 ] Crit: 7  Depth: 3
ID:  10 XYS: [ 4 , 2 , 7 ] Crit: 8  Depth: 4
ID:  11 XYS: [ 4 , 2 , 6 ] Crit: 9  Depth: 4
```

Figure 27. Construction Blueprint Simulation

A depth of 0 is assigned to all blocks on the floor, with the depth increasing for the amount of blocks that must be placed prior to itself - which we call the construction step. The benefit of these steps is that it allows multiple agents to construct a larger structure at once by requiring all blocks of the same depth to be placed before the next step can take place. The "Crit" or critical block is a single block that it is dependent on, which is used to assign orientation further on during processing. During this step, validation that no floating blocks exist in the goal structure also occurs. The orientation and block type are then assigned from the critical blocks and construction step data.

3.5.4 Blueprint Data

The data generated by the algorithm needs to be useful for developing the blueprint that the blocks and inchworm will use for construction. To enable a decentralized building process, each block needs information it can pass to other blocks or the inchworm when it is picked up. The data required for each block is as follows:

1. Coordinates for placement
2. Orientation of the peg for strong linked blocks
3. Rotation of the block face that accepts the next peg
4. Coordinates for the next block to be placed

Once a block is picked up for the first time by the inchworm, the coordinates for placement, orientation, and rotation are transferred to the inchworm where it will be ferried. Once the block is placed and validates its location, the coordinates for the next block are transferred to the inchworm and the inchworm releases this block. Following this design, we are able to develop a blueprint that is distributed and stored on the blocks for the inchworm to communicate with, removing the need for an external director once construction begins.

Chapter 4: Experimental Evaluation

4.1 Smart Block System

4.1.1 Block Unit Testing

We performed unit tests for the smart blocks' communication system after the blocks were fully implemented. Since the smart block design does not have a means of visual indication of its behavior, a headless end-effector module (referred to as the "EE tool") was created to flash colored LEDs upon the correct interaction and communication with the blocks. The goal of unit testing for the smart blocks was to verify the behavior of the blocks (and the associated communication pipelines) under different construction conditions. These conditions were: initial block pickup, pickup and placement of a block that is connected to one other block, and pickup and placement of a block connected to two other blocks.

The first series of tests involved a single smart block and the EE tool to test baseline functionality for block-to-robot communication. The block was placed into its before-pickup state and the "correct" side was set to the right-facing side as shown in Figure 28. The white LED on the EE tool was instructed to only flash when the EE tool was connected to the correct side and received a confirmation signal from the block. When placed on any other face of the block, the white LED was to remain off.

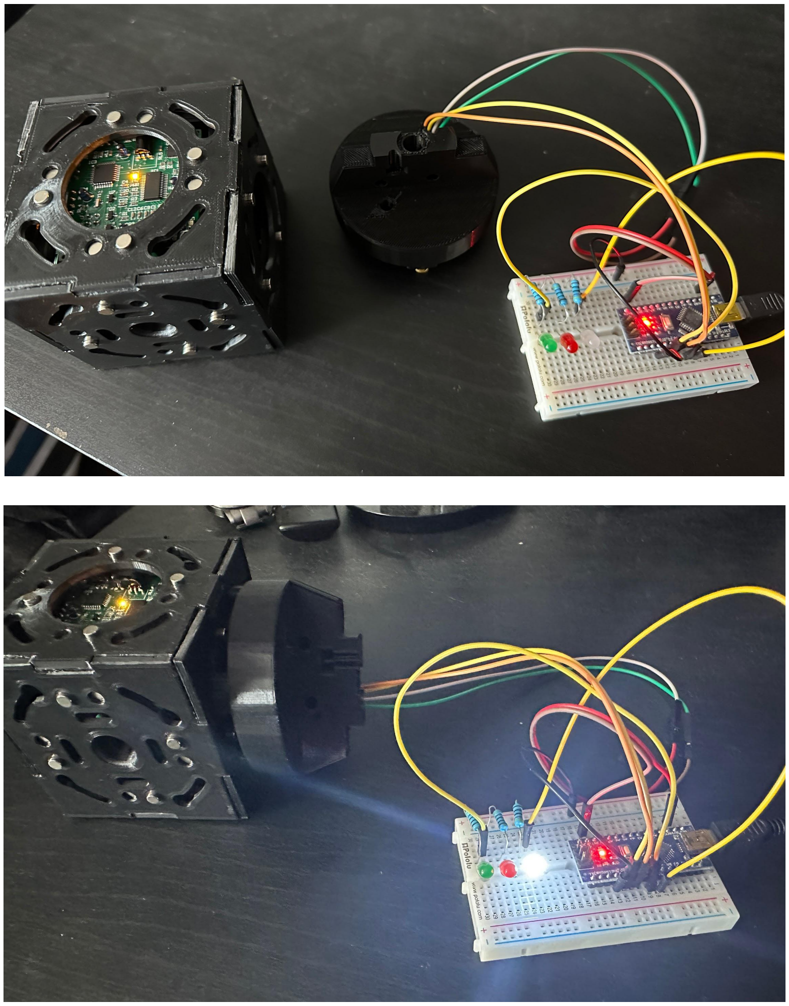


Figure 28. Before (top) and after (bottom) connecting the EE tool to the “correct” face of a block.

The next series of unit tests involved two interconnected smart blocks in determined orientations, with the second block connected to the EE tool set in a state reflective of just having been placed. The setup is shown in Figure 29. The tests were run with the orientation condition checks and the same “correct face” condition for the EE. Further, the second block was to check with the first block to verify that it is at the correct location. The green LED on the EE tool was to flash only when all these conditions were met.

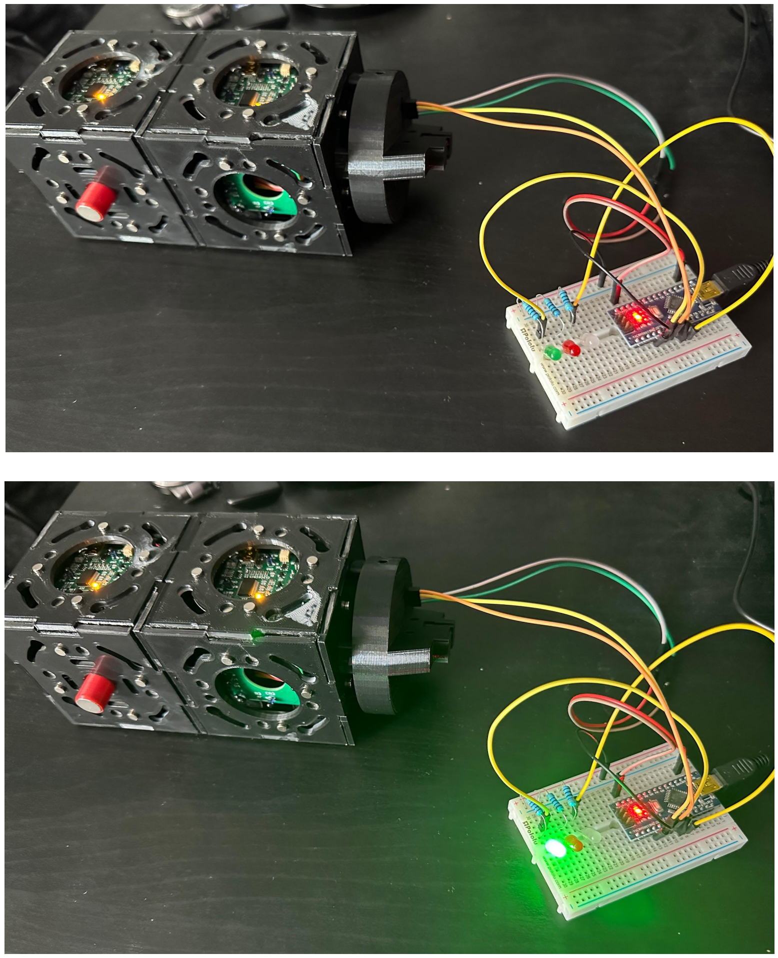


Figure 29. Before (top) and after (bottom) “placing” the second block with the EE tool.

The same series of tests were repeated with the only change being the EE tool emulating the placement of a third block. This test is shown in Figure 30. The success of the individual trials across all three unit tests were recorded in Table 6 in Appendix C.

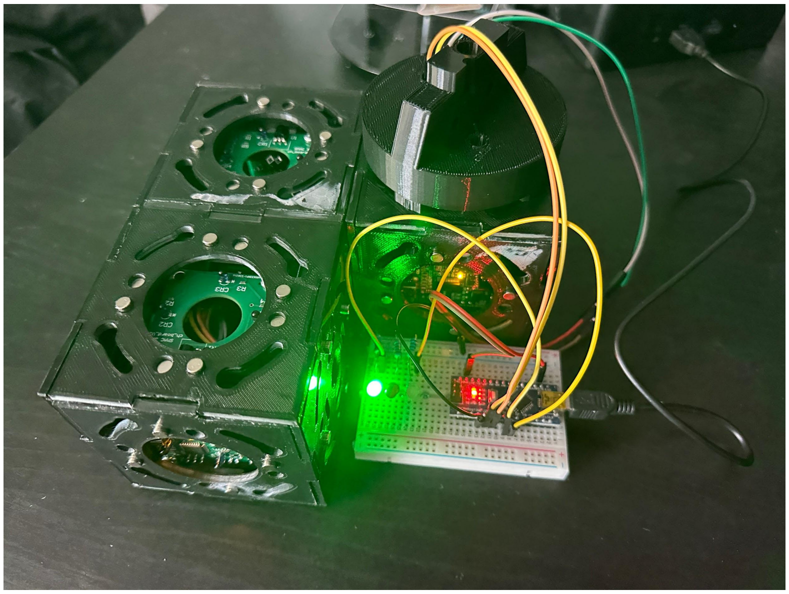


Figure 30. Unit test after “placing” the third block with the EE tool.

4.1.2 Block Unit Testing Analysis

Through the unit tests performed on the blocks to emulate block-to-block communication and block-to-robot communication, we can gather that the system works as intended. The unit tests evaluated block behavior for basic construction operations, including: block pickup, placement of a block that is connected to one other block, and placement of a block connected to two other blocks. These operations are integral to any robotic construction system.

By emulating these behaviors in the blocks, these unit tests serve as a proof-of-concept of our stigmergic block behavior. For a heterogenous block design, building a structure depends upon compliance with a few important conditions:

- the robot picks up the correct block - aside from the importance in order of construction, specific types of blocks must be placed since we have two types of blocks (90° and 180°);
- the robot places the block in the correct location - like any construction process, the right materials need to go in the right places according to a plan;
- and the block is in the correct orientation - we are trying to build strong and stable structures by optimizing the strong links, and therefore, orientation is integral.

The success of the first series of unit tests proved that our block-to-robot communication works, and demonstrated the end-effector’s identification of a specific smart block and block face.

While an inchworm would pick up a block from its top-most face, we choose to verify a "correct" face because the top-most face of a block is dependent on its orientation.

For the second series of unit tests, our success indicated the fulfillment of both the first and second condition mentioned earlier as well as block-to-block communication. These tests show that the proper behavior occurs when adding a block to an existing structure, even if that structure is just one other block. Further, these tests indicate the success of the orientation and placement checks upon adding this second block.

Lastly, the third series of unit tests were successful, further reinforcing the observations made for the second series of unit tests. By testing the addition of a third block, we demonstrate the smart block system works beyond just two blocks and this indicates functionality for a multi-block system. After the three unit tests succeeded, we can consider our smart block system as compliant with the defined conditions that allow our system to build a structure. Thus, these tests exhibit the smart block system's stigmergic behavior and their ability to intelligently create structures.

4.2 Blueprint Structure and Stability Validation

4.2.1 Weight Testing

Weight testing involves ensuring that no block will exceed a specified weight limit which is configurable in the simulation. This is to ensure a single block is not under the pressure of dozens of blocks which could cause it to fail. The weight limit for these blocks is set at a conservative 15 additional blocks to ensure at no point will a block reach critical failure.

The weight value of each block is determined by finding all blocks that are dependent on that block and adding up the total weight. This means for a block on the ground, all blocks that rely on that block being placed prior to it will add to the weight value. Completed testing of the weight limit is displayed in Figure 31, where multiple structures held with a single block will fail the weight testing. The weight can be corrected by adding more connections to ground as seen on the right. The weight value only needs to be calculated once for the entire structure because the weight will only increase as blocks are placed, so we just test the maximum weight.

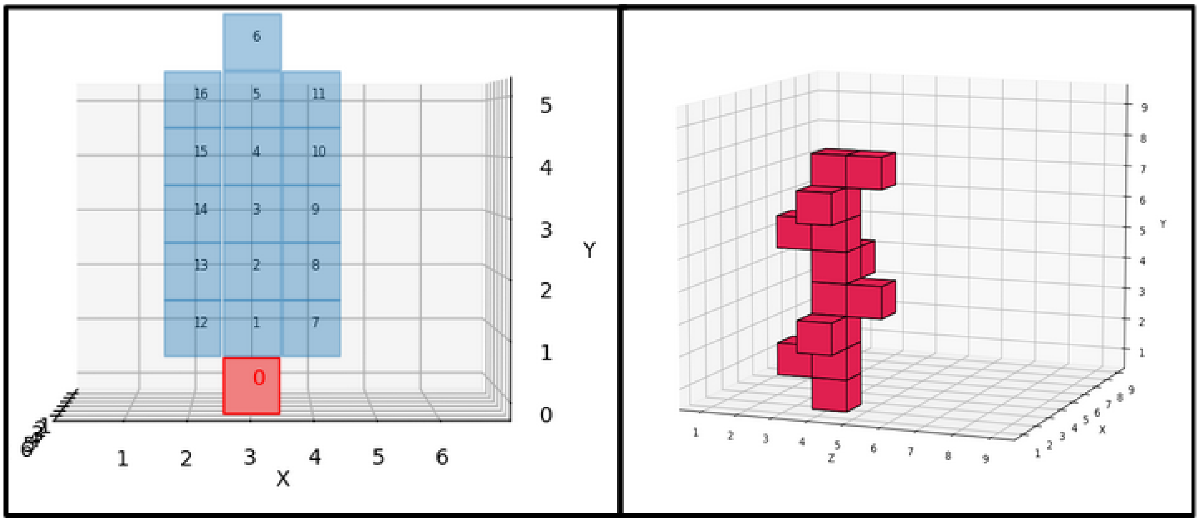


Figure 31. Left: Structure failing weight validation with 16 blocks all supported by one block. Right: A passing structure with 14 held on one block.

4.2.2 Stability Testing

A goal structure is only possible if it remains stable during and after construction. To verify this for any structure, validation of structure weight and overhang blocks is done in an iterative process. The algorithm runs a simulation of construction and tests the stability of the structure as each block is placed. If a block is placed that exceeds the physical constraints of the connection, the process is terminated and the user is notified of exactly where the failure would occur during simulated construction with an iterative visual shown in Figure 32. This allows the user to make edits to the structure at the simulation step before construction begins. Unlike previous iterations of this project, the strong joints of the new block design prioritize vertical building until an overhang occurs, at which horizontal strong points are required.

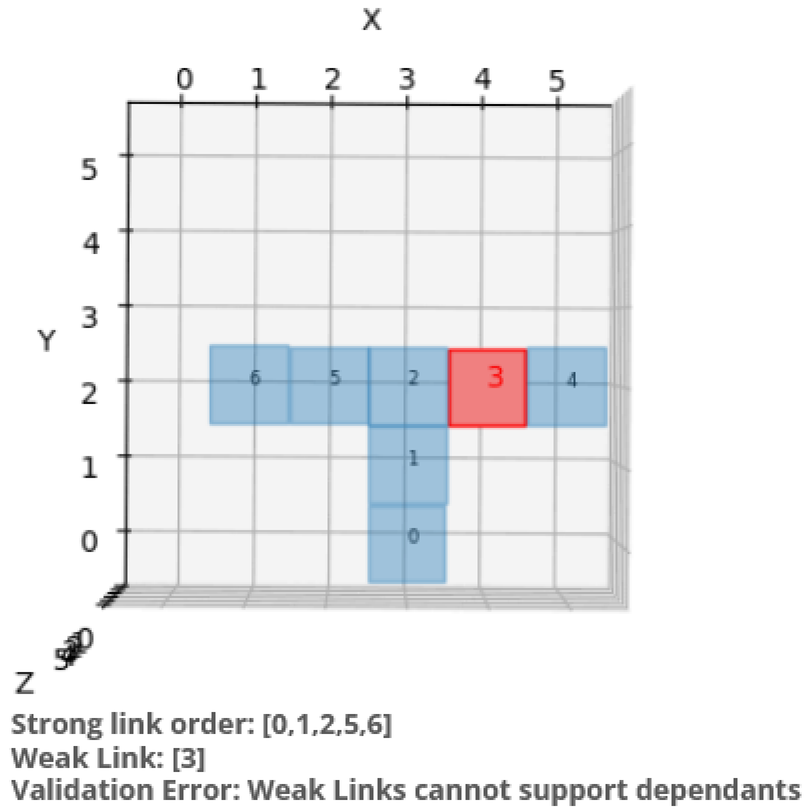


Figure 32. Visual of validation failure graphic.

Blocks are first classified as one of six possible types as they are placed. Each block can be classified as the following: Floor, Pillar, Corner, Overhang, Hanging, or Stacked on Overhang. This is to simplify the logic for each block based on its placement in the structure.

The stability algorithm prioritizes blocks that need strong links the most, then weak links other dependents if necessary. This prioritization is done on a heuristic using the weight value, block classification, depth, and critical values. Constraints used for this project were allowing overhangs of 3 blocks maximum that are linked with pegs, as the weight of the blocks with the robot could prove too heavy for the base block. The simulation allows for blocks not linked with pegs, but those blocks cannot have dependent blocks.

This simulation has the capability to simulate construction and validate hundreds of blocks efficiently as seen in Figure 33 so that structures can be tested before real-world implementation occurs. Structures with up to 1000 blocks can be processed and validated with a time complexity within the constraints defined in Chapter 3.5.1, with complete structures being processed in under 500ms.

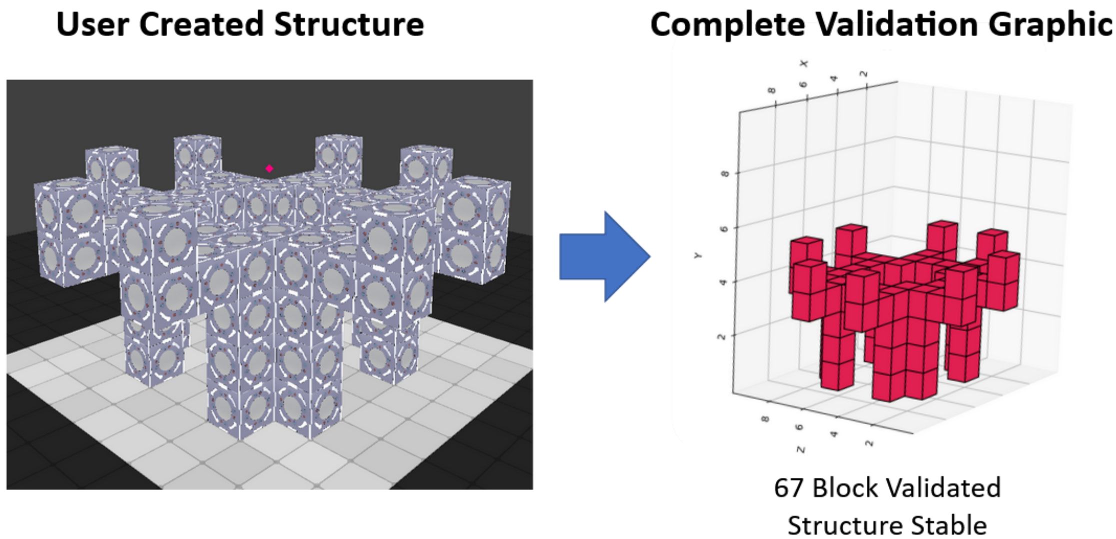


Figure 33. Validation of 135 block structure.

There are still some limitations in the prioritization heuristic of the algorithm. The algorithm may fail on structures that could be built by the platform, but the algorithm incorrectly prioritized the strong link.

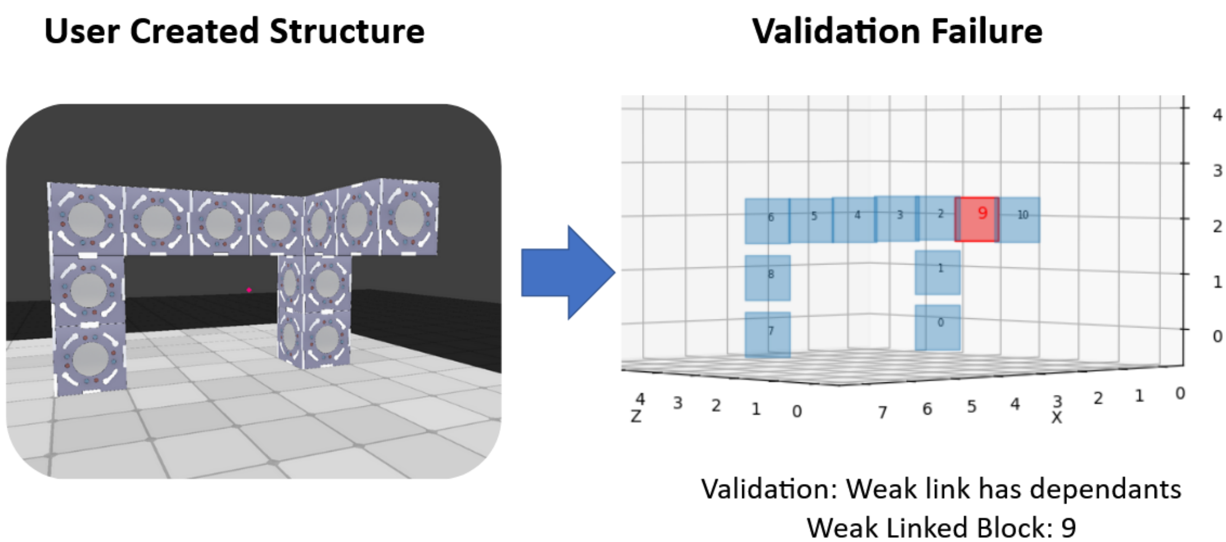


Figure 34. Validation failure on a physically-stable structure.

In Figure 34, block #2 is a corner block with two overhanging neighbors. The algorithm decided to strong link with block #3, because it has just as many dependants as block #9. As a result, block #9 is weak-linked and fails validation because it also has dependents. However, this structure is actually possible in reality, because block #6 can handle blocks #5, #4, and #3 as a single overhang, allowing block #2 to strong-link to block #9. The algorithm fails this structure because of how it chose to place the strong links. In Figure 35 is an example of the algorithm passing this same structure in a different orientation with respect to the base frame.

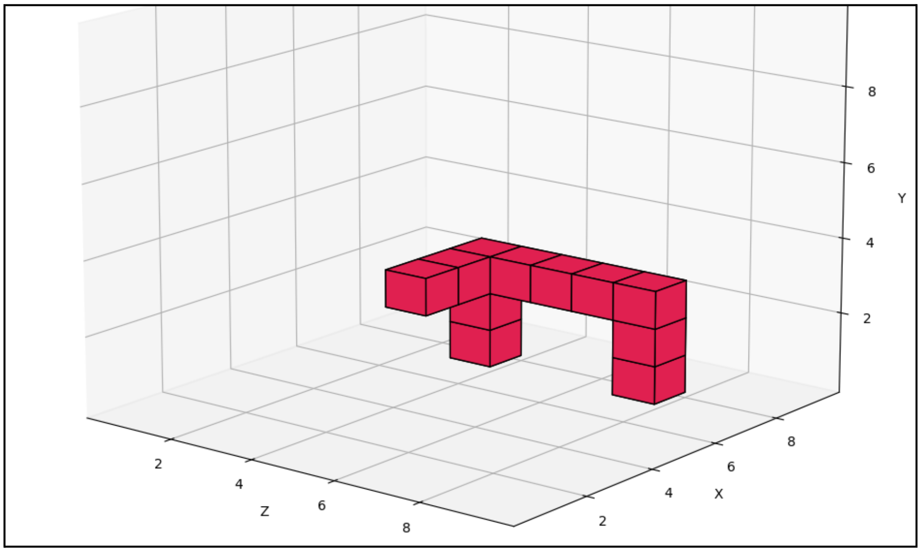


Figure 35. Validation success on the same physically-stable structure.

In the completed graph, block #3 is weak linked, and blocks #4 and #5 and strong linked to block #6. This allows a strong link from block #2 to block #9 and does not break any constraints of the algorithm. With a different orientation of strong links, the algorithm is able to create a passing structure blueprint.

4.2.3 Blueprint Generation

The data from the simulation can be exported as a plain text document with all information needed to generate a blueprint for the blocks and inchworm to autonomy create a structure. The blueprint is only exported if the entire structure is validated with no errors. However, this does remove the option for real world failure testing.

Each block requires information to transfer to the inchworm for placement. We have defined the information needed for placement with the following syntax:

{[x,y,z]:orientation:rotation:[x,y,z]:[neighbors]}

For every block, it must know the position it needs to be placed [x,y,z], the orientation and rotation of the block for peg placement, the next block to navigate to, and a list of this block's neighbors for validation once it is placed. The neighbors is designed as a list of [x,y,z] coordinates. The address of each block is simply the number it is in the list. A total of 7 bytes is needed for all the information before neighbors, with a total of 3 additional bytes for a maximum of 27 total bytes per block including its address. Figure 36 displays a validated structure and the subsequent blueprint generated by the algorithm.

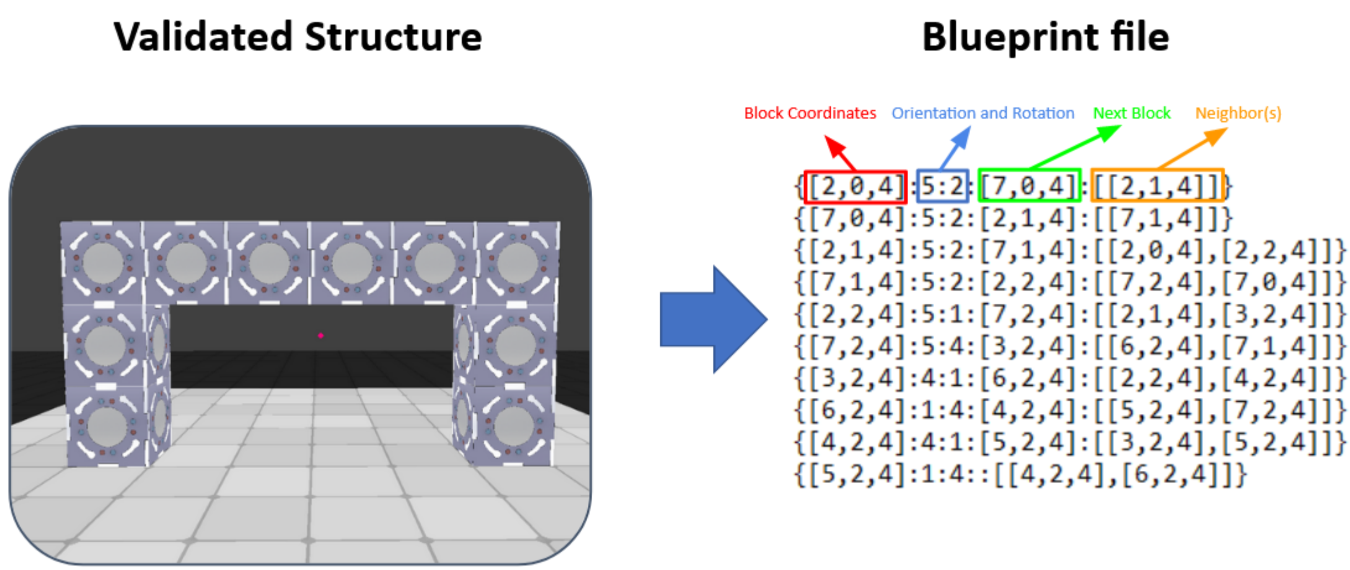


Figure 36. Validated block structure with corresponding blueprint.

While the blueprint interpreter is not within the scope of this project, the plain text file conceptualizes the blueprint needed for each block and can be scaled based on the amount of data capable of being stored on each block. The interpreter would take these messages and create transmissions using the protocol outlined in Figure 21.

4.2.4 Algorithm Discussion

Each previous iteration of this platform has developed some type of simulation and algorithm for their stigmergic building design. SMAC 1.0 developed an algorithm that could handle multiple inchworms with homogeneous blocks. MARIA's algorithm was able to conduct stability testing and handle weight requirements. In comparison, this algorithm is able to conduct block, stability, and weight validation on heterogeneous blocks and develop a blueprint for construction.

In this iteration, the algorithm did not meet the standard definition of *completeness* [12] but it did achieve the level of completeness defined in Chapter 3.5.1. If a blueprint is generated for a structure, it is guaranteed to be buildable by the smart block and inchworm system. However, due to cases noted in Figure 34 and Figure 35, a solution that may exist is not always found. This was due to a design decision to avoid checking every possible orientation which does not meet our time complexity goal on large structures. There are possibilities to adapt this algorithm to become complete by adding a method that could "guess and check" different block orientations to find the *optimal* [12] orientation of each block, but we chose for this simpler design as the first iteration of a heterogeneous block blueprint generator.

We focused on completing the algorithm to handle the peg block design, so it is built for use with a single inchworm. However, the depth data attribute that is generated from this algorithm allows multiple blocks to be placed at the same time, for instance the two bases of a bridge.

Unlike all previous iterations, this algorithm handles prioritization and selection of peg orientation for each block in a structure. It also has a custom simulation interface that allows users of the system to easily design their own structures. The biggest iteration over previous algorithms is the ability to export a structure's data to a blueprint that can be used by the blocks and inchworm for construction.

There are still improvements to be made to the algorithm. In Chapter 4.2.2, a structure that could be built had improper peg orientation built, causing it to fail validation. The algorithm's orientation heuristic focuses on prioritizing the heaviest overhangs and dependents, and does not test multiple peg placement options in one pass. This design choice was made to limit the time complexity of processing large structures where hundreds of possible blueprints exist.

Chapter 5: Conclusions

5.1 Summary

In this paper, we have presented a novel block mating system iterating on the smart block design created by the MARIA team, along with a complete algorithm for the new system. The new block design increased overall tensile and shear loads compared to previous iterations and allowed for larger overhangs with support for the inchworm. The mechanical testing yielded a 53% increase in tensile loading and a 330% increase in shear loading. The new algorithm allows any custom structure to be built by a user in a simulated environment, followed by complete validation of every block and iterative construction testing under weight and stability constraints. With the new validation algorithm, the system can guarantee a given structure can be built or report the failure node visually to the user.

5.2 Future Work

5.2.1 Removable Peg Placement

For the scope of our project, the team decided not to solve the problem of peg placement, and instead for our demonstrations, the pegs were manually manipulated. This decision was made early on in the project due to limitations on the time and budget required to create a "peg-dispensing mechanism". However, future iterations of the project could design and implement a mechanism that feeds these pegs to the end effector when a strong-link must be placed (should a peg-based system be continued). Tackling this challenge would bring this proof-of-concept system closer to a real-world implementation.

5.2.2 Mating System

SMAC 3.0 explored a heterogenous block design that prioritized strength and stability within structures of multiple blocks, while not significantly compromising modularity of the system. We recommend that future work on the SMAC platform continue working towards making "the perfect block" that is easy to manipulate and also capable of building non-trivial structures. Ideas for exploration might include using mechanical actuation within blocks to create strong links or electro-permanent magnets to explicitly control the connections between blocks.

Further, the SMAC 3.0 block system specifically explored the constraint of having two strong-links per block at either 90 or 180 orientations. Future work may find it valuable to explore variations of this constraint, such as having three-strong links per block, which could enable more complex backbone chains and even stronger structures.

5.2.3 Design For Manufacturability

The manufacturing process of our block prototype has significant room for enhancement because it involves assembling numerous individual pieces like magnets with superglue, which does not provide a permanent bond. This led to some magnets falling out of place over time due to the magnetic strength overpowering the glued connections. Manufacturing these blocks was very time-consuming and did not allow for ease of access to the inner components of the blocks after completing the building process of an individual block. This issue can be tackled by adding access holes at strategic positions or even creating a latch mechanism for one face per block that will allow access within the core for maintenance while maintaining a tight seal on said face when needed. The ease-of-access for the microcontroller, battery, and switch are all important when working with the smart blocks. An additional aspect that requires improvement in the block prototypes is utilizing more authentic building materials that would make the blocks more robust.

On the electrical side, custom PCBs were designed for the IR breakout boards and the microcontroller boards. They were sent to a commercial manufacturer to produce and assemble most of the components on the boards. While it was possible to manufacture these PCBs with university resources, the high quantity required for several smart blocks made it infeasible given the expertise among the team and timeframe. As such, a major expense for the team was allocated to JLCPCB for their service and parts from their turn-key supplier. Future teams with a dedicated ECE may consider redesigning the microcontroller board to use a smaller, cheaper generic microcontroller module with castellated pads so that it may be soldered on top of a breakout board with other peripherals - which would be, in turn, much cheaper and simpler to manufacture.

5.2.4 Fully Complete Algorithm

The blueprint generation algorithm meets our criteria for completeness, but does not always find a solution where one may be possible. Future iterations of this algorithm can add capabilities to check alternative block orientations for a single structure to find solutions outside of what the current heuristic optimizes for.

5.2.5 Blueprint Interpreter

The current implementation of the algorithm and simulation develops a plain-text blueprint with all the information a system needs for stigmergic construction. The next step in this process is creating an interpreter that parses this information and encodes the messages into IR messages to be propagated among all blocks.

5.3 Lessons Learned

In developing the new block design and complete algorithm concept, we took inspiration from the previous projects and goals they have achieved as benchmarks for us. However, we found it easier to build the blocks and algorithms from the ground up to understand each component at a lower level. Over the course of the project, we learned that planning only goes so far in development and no plan goes perfectly the first time. Our expectations for portions of this project were different from reality, meaning what was supposed to work the first time actually required substantial revisions.

Given the opportunity to start again, we would push to build a prototype, even a basic one, within the first two months. Delaying the first prototype only meant all the iterations after had to come faster and better, putting strain on our project. Running into issues or failing is good, but should happen early in the project. We learned that multiple expectations are a necessity, and problems you never could foresee in simulation will arise with a physical prototype.

References

[1] “Changes and Advancements in the Construction Industry,” *The Cat Rental Store Blog*. [Online]. Available: https://www.catrentalstore.com/en_US/blog/changes-advances-in-construction.html. [Accessed: Apr. 26. 2023].

[2] “Building Construction,” CC Corp. [Online]. Available: https://www.contourcrafting.com/building-construction. [Accessed: Apr. 26. 2023].

[3] A. Khuong et al., “Stigmergic construction and topochemical information shape ant nest architecture,” in *Proc. of the National Academy of Sciences*, vol. 113, no. 5, pp. 1303–1308, Jan. 2016. [Online]. Available: doi, https://doi.org/10.1073/pnas.1509829113. [Accessed: Apr. 26, 2023].

[4] R. Nagpal, K. Petersen, and J. Werfel, “TERMES: An autonomous robotic system for three-dimensional collective construction,” in *Proc. of International Conference on Robotics Science and Systems, 2011*. [Online]. Available: doi, https://doi.org/10.15607/rss.2011.vii.035. [Accessed: Nov. 8, 2022].

[5] B. Jenett and K. Cheung, “BILL-E: Robotic Platform for Locomotion and Manipulation of Lightweight Space Structures,” in *Proc. of AIAA/AHS Adaptive Structures Conference, 2017*. [Online]. Available: https://doi.org/10.2514/6.2017-1876. [Accessed: Nov. 1, 2022].

[6] M. Allwright, N. Bhalla, H. El-Faham, A. Antoun, C. Pinciroli, and M. Dorigo, “SRoCS: Leveraging Stigmergy on a Multi-robot Construction Platform for Unknown Environments,” in *Proc. of International Conference on Swarm Intelligence, 2014*. [Online]. Available: doi, https://doi.org/10.1007/978-3-319-09952-1_14. [Accessed: Nov. 1, 2022].

[7] P. Dench, S. Kalimani, B. Waid, E. Torregrosa, M. Charter, and M. Mohapatra, “MARIA: Multi-Agent Robotic Intelligent Assembly,” Major Qualifying Project, Worcester Polytechnic Institute, Worcester, MA, May 05, 2021. [Online]. Available: https://digital.wpi.edu/pdfviewer/m039k804v. [Accessed: Nov. 1, 2022].

[8] D. Brandt, David Johan Christensen, and H. Lund, “ATRON Robots: Versatility from Self-Reconfigurable Modules,” in *Proc. of International Conference on Mechatronics and Automation, 2007*. [Online]. Available: doi, https://doi.org/10.1109/icma.2007.4303511. [Accessed: Nov. 1, 2022].

[9] C. Collins, J. Contreras, N. Dhanaraj, H. Liang, T. Rizzo, and C. Wagner, “SMAC: Symbiotic Multi-Agent Construction,” in *Proc. of IEEE Robotics and Automation Letters 6.2, 2021.* [Online]. Available: doi, https://doi.org/10.48550/arXiv.2010.08473. [Accessed: Nov. 1, 2022].

[10] E. Benevedes, D. Ferro, F. Kernan, and S. Gould, “RAINSTORM: Rooftop Assembly Inchworm Network and Swarm Tiling Optimization for Rooftop Maintenance,” Major Qualifying Project, Worcester Polytechnic Institute, Worcester, MA, Apr. 28, 2022. [Online]. Available: https://digital.wpi.edu/pdfviewer/xw42nc13. [Accessed: Nov. 1, 2022].

[11] “NEC Infrared Transmission Protocol,” Altium, Sept. 13, 2017. [Online]. Available: https://techdocs.altium.com/display/FPGA/NEC+Infrared+Transmission+Protocol. [Accessed: Dec. 10, 2022]

[12] A. Mackworth, “CPSC 322, Practice Exercise Solutions to Uninformed Search,” UBC Computer Science 322: Introduction to Artificial Intelligence, 2013. [Online]. Available: https://www.cs.ubc.ca/~mack/CS322/exercises/1_ex_search_uninformed_sol.pdf. [Accessed: Apr. 26, 2023].

[13] “ursina engine: open source game engine,” ursinaengine.org. [Online]. Available: https://www.ursinaengine.org/. [Accessed: Apr. 26, 2023].

Appendices

Appendix A

Simulation and algorithm github: https://github.com/jleavitt-git/SMAC_3
Motion planning github: https://github.com/krlepping/SMAC_3.0

Appendix B

The testing setup for initial IR communication as described in Chapter 3.4.1 is shown in Figure 37.

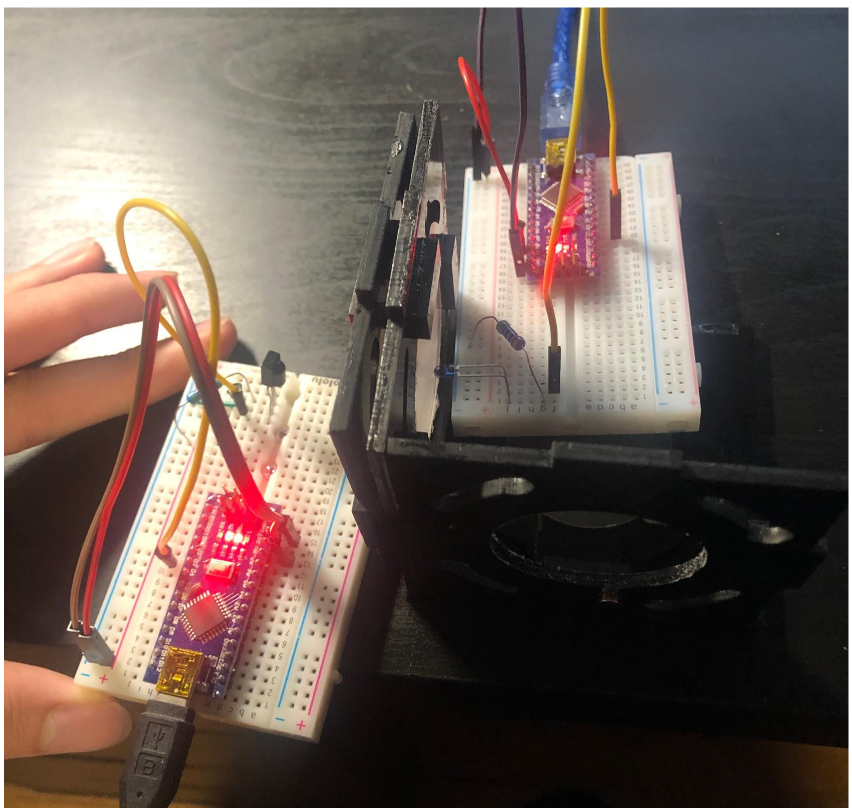


Figure 37. Preliminary IR communication tests.

Appendix C

Table 6 shows the data for the unit tests described in Chapter 4.1.1.

Table 6. Block Unit Testing Trials

Trial	Unit Test 1	Unit Test 2	Unit Test 3
1	TRUE	TRUE	TRUE
2	TRUE	FALSE	TRUE
3	TRUE	TRUE	TRUE
4	FALSE	TRUE	TRUE
5	TRUE	TRUE	TRUE

Observations/Notes: Trial 2 of Unit Test 2 failed due to the second block "catching" an erroneous signal from a different face on the first block.

Appendix D

An 8x6 (block units) baseplate was created to help test our autonomous robotic construction system, as shown in Figure 38. This was designed using 0.125" acrylic sheets that were laser cut to form a "box" that fits over a MDF panel (not pictured). The underlying MDF panel has strips of mild steel positioned along the length of the baseplate and under each peg hole, such that our magnetic pegs can lock into the baseplate.

Figure 38. SMAC 3.0 baseplate.