Spike: A Novel Cube-Based Robotic Platform

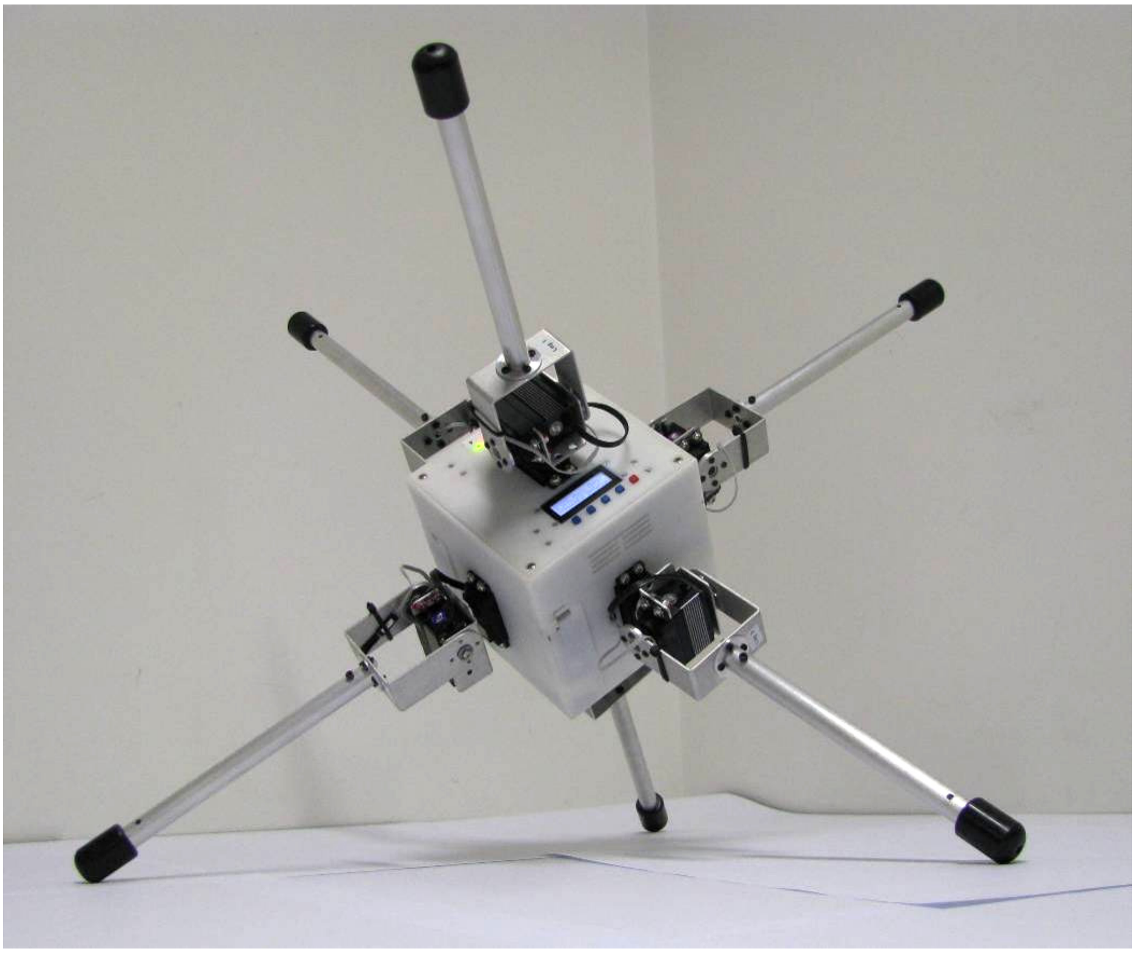
Christopher Keith Coyte

A thesis submitted to
Auckland University of Technology
In fulfilment of the requirements for the degree of
Master of Engineering (ME)

2010

School of Engineering

Primary Supervisor: Dr. John Collins



The ‘Spike’ Robot

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ATTESTATION OF AUTHORSHIP

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been submitted for the award of and other degree or diploma of a university or other institution of higher learning.

Christopher Keith Coyte

ACKNOWLEDGEMENTS

I would like to acknowledge several people for their technical assistance and support.

For the successful completion of my master's thesis, the robotic development required many aspects of engineering to unite, involving technical skills beyond my chosen electrical specialisation in engineering. My length of time at university has allowed me to integrate with the faculty and to be fortunate enough to work with some of the most brilliant minds at this university, with ideas and knowledge that have assisted inspiration of this research, and achieve successful outcomes.

I wish to sincerely thank both of my project supervisors John Collins and Mark Beckerleg for their invaluable advice, guidance and assistance in research and development, providing technical expertise on numerous aspects over the entire research and development process, combined with motivational encouragement during all development stages.

I would also like to thank Hans Oberst and the Creative Industries Research Institute at the Auckland University of Technology for assistance with SolidWorks CAD design and fabrication. The combination of efforts and advice that has been provided by the university technicians was invaluable in achieving a successful design and fabrication.

Finally, and most importantly, I would like to thank my parents for their continued support both financially and morally, assisting me throughout my entire education. They have provided drive and a pool of resources that have aided me to persevere with my education.

DEDICATION

I wish to dedicate my work in this thesis to my parents Bryan and Sue Coyte. This master's thesis represents my greatest effort of academic achievement, and a life long fascination with making, breaking and fixing all things engineering related. My parents have assisted by creating situations where I have been able to explore and develop my interests and skills in numerous environments. Without the resources and opportunities that my parents have assisted me with, I would not have developed my own abilities, both academically and in the working environment, to the extent that I now have.

ABSTRACT

This thesis documents the successful development of a novel form of robotic motion, inspired partially from bio-mimicry and partially from a search for unknown forms of legged robotic motion. The novel motion developed from the research in this thesis, was created by implementing a tilting and falling action on the robot as a result of its leg movements to form a unique robotic gait. The tumbling gait was developed on a six legged robot that featured a tri-pedal stance, with the robot driving its legs in such a way as to unbalance the robot and force it to fall onto an opposing leg, creating a tumbling motion.

To create the desired tumbling motion, several robotic concepts were investigated. The final robotic version used a series of leg actuations that induced a tumbling action on the robot. This was achieved by moving one of the three grounded legs towards the centre of the robot, causing the robot to tumble in that direction, giving a triangular footprint to the motion.

After determining the method most suitable for motion and reviewing the current literature in this field, the robotic chassis design was created based around the manufacturing abilities of a plastics three dimensional (3D) rapid-prototyping printer. This design was cube-shaped with a single leg protruding from each face of the cube, giving a total of six legs. This cube design included three axes of symmetry, featuring a collinear pair of legs mounted on each of the robot’s axes. The cube design incorporated leg mounting, electronic board mounts, battery compartments and internal strengthening.

Due to the complexity of the design and the high level of integration between the electronics and mechanics, ‘off-the-shelf’ components could not be used, and a complete electronics design was created, prototyped and built, comprising multiple circuit boards for the required power, processors and input-output devices. Considerable electronic development permitted a wireless controlled semi-autonomous robot mobilised by the combined use of twelve servo motors for evaluation of its gait. The electronic design also facilitated interchangeable electrical systems for expansion and continued development of its control and motion.

An initial direction for the robot to follow was provided by a wireless controller to the onboard navigation system. The navigation system then performed mathematical

calculations based on the robot's triangular footprint, to determine which leg to move, in order to proceed in the required direction. The tri-pedal stance inherent in the robot's footprint, allowed each of the robot's steps to be directed toward one of three directions. It was found that the accuracy of following a given heading over time was dependant on the number of steps taken by the robot. The manually programmed leg movements for Spike's gait were generated from a lookup table, and programmed into the robot's processors, relating servo motor angular displacement to leg destination position for each orientation of the robot.

Spike represents a true example of a simple, but effective concept of motion, which has been crafted in the form of an elegant mechatronics design. The motion of the robot is unique by its simplicity of concept, but may be adapted for development of complicated gaits. It is intended that future expansion of electronic control will provide added research potential, through the development of new sensors and systems, and experimentation with evolved controllers that may subsequently be evaluated on the robot. The final result of this thesis lies with the custom designed six legged robot that successfully implements a unique and novel form of locomotion.

CONSEQUENTIAL RESEARCH OUTPUTS

During the undertaking of this thesis, the writer had the following research outputs:

Publications

Coyte, C. "Spike: A Six Legged Cube Style Robot". Second International Conference on Intelligent Robotics and Applications. ISBN: 978-3-642-10816-7 LNAI: 5928 pp535-544 ICIRA2009 Singapore. Springer-Verlag Berlin Heidelberg (2009)

Oral Presentations

Coyte, C. Spike: A Six Legged Cube Style Robot. School of Electrical and Mechanical Engineering Seminar Series. Auckland, New Zealand: AUT University. (2009)

Coyte, C. "Spike: A Six Legged Cube Style Robot". 2010 Embedded Systems Workshop. Auckland, New Zealand: Manukau Institute of Technology. (2010)

1 INTRODUCTION

1.1 Overview

This thesis involves research to determine a new and novel method of robotic motion for legged mobile robotics. This development was achieved with the successful creation of a uniquely designed robot, fabricated specifically to facilitate a creative and novel form of motion. The uniquely designed physical robot is a platform for robotic motion research, and allows experimentation and analysis to be performed in real world environments. The work in this thesis has contributed towards a paper titled "Spike: A Six Legged Cube Style Robot" published in the "Proceedings of the 2nd International Conference on Intelligent Robotics and Applications" [1].

In this thesis, the legged robot named 'Spike' was designed and fabricated for the purpose of delivering the envisaged motion concept, inspired in part from bio-mimicry, and from a desire to discover new forms of motion. The constructed robot is shown in Figure 1-1.

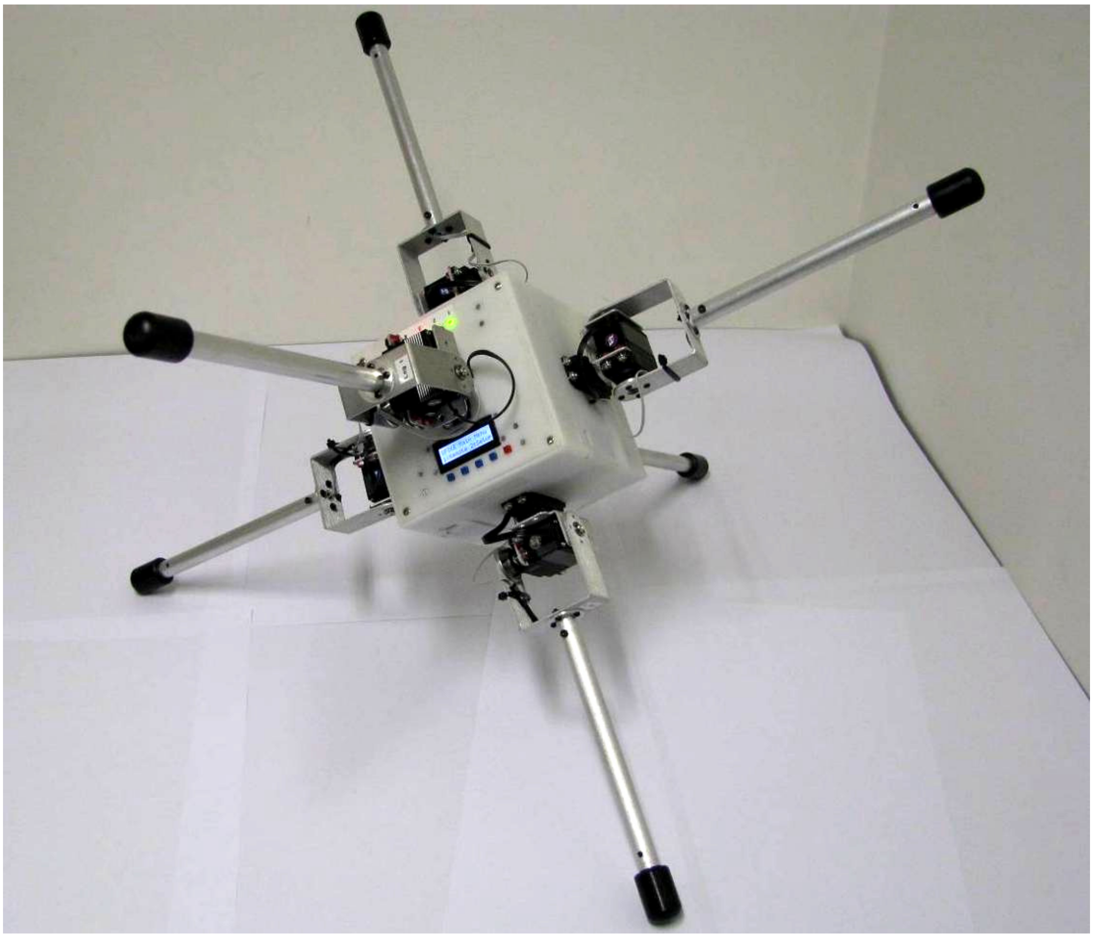


Figure 1-1: 'Spike' the Six Legged Robot

The creative method for motion consisted of a series of leg actuations to induce a tumbling and falling action, creating the robot's motion. The principle of this motion relied on the creation of a weight imbalance, combined with the use of friction and gravity to create a tumbling motion allowing the robot to move. To create the tumbling motion, a novel concept of leg movement was developed, and consequently a robotic framework was designed and constructed.

The central framework of the robot consisted of a cube shaped chassis, designed and fabricated on a plastics 3D rapid prototyping printer. The robot used a total of six legs, one fitted to each face of its cube shaped body. Each leg acted independently from the other legs, driven by electromechanical servo motors to induce the required tumbling actions. The cube shaped body was designed with the intention of enclosing all electronic circuit boards, custom designed specifically to integrate within the chassis's central cube space. The robot's name 'Spike' was subsequently derived from its appearance.

The configuration of the legs around the cube allowed the robot to take a tripedal stance when placed on the ground. Light sensors mounted within each foot of the robot's six legs allowed the robot to determine which three of its six legs were in contact with the ground surface at any given time. The robot's motion was generated by directing a single supporting leg towards the middle of the opposing side of the triangle formed by the three supporting legs. By shifting the legs using this method, the robot's centre of mass was forced toward a tipping fulcrum, created by the two remaining standing legs. At the position where the centre of mass extended past the fulcrum point, the robot became unbalanced, and tumbled into a new tripedal leg configuration and orientation.

Spike was programmed to follow a user defined heading: an angle relative to the robot's starting position, transmitted to it by a wireless remote controller. However the robot could not directly do this, as at any one time it could only move in one of three possible directions due to its triangular footprint. The robot travelled by taking successive tumbling steps in the closest direction toward the given heading that the tripedal stance allowed. After each tumble, the robot was required to determine its new orientation, with respect to its original defined heading, and dynamically reconfigure its planned motion to achieve navigation toward the given target heading. After several steps, directional errors accumulated, and the robot was required to recalculate its trajectory to compensate for these directional errors. In this way, the accuracy of the overall motion improved with the increased number of steps taken. The robot continued to shift in the user's given direction without intervention, until the user selected termination of the robot motion. As the robot moved, relevant data was transmitted continuously from the robot to a nearby desktop PC for analysis.

The development of the robotic hardware required several aspects of engineering to be successfully integrated. The design process consisted of: motion concept planning, mechanical development, electrical development, construction, programming and evaluation. Through investigation of many current forms of robotic motion, it was determined that this type of motion, using a robotic design such as Spike's, had not previously been constructed.

It is planned that the robot will be available for further research on evaluating evolutionary computation based control systems for the robot's motion, with an objective of finding new gaits, that are able to move differently, and perhaps more efficiently than the manually programmed gait developed from this research.

1.2 Thesis Aim

The principle aim of this research was to create a novel form of robotic motion. The process of creating this motion required many creative ideas to be explored and analysed for their ability and effectiveness. With many ideas considered for creating motion, the selected method inspired the design and construction of the novel robotic chassis. The methodology of creating motion is discussed in detail in section 3 of this thesis.

The initial planning of the cube-based robotic design involved a basic analysis of a simplified robotic model, with consideration of how the envisaged method of creating motion could be realised. The process of discovering and planning how the robot might move to overcome its inherent stability represented a challenge that was overcome in this development, with highly skilled engineering and a meticulous attention to detail. Determining the originality of the robotic design and motion was performed by an in-depth review of robots, with specific reference to robotic motion and novel design.

After construction of a robotic framework consisting of both mechanical and electrical design, adequate programming of the multiprocessor robot led to a fully operational gait.

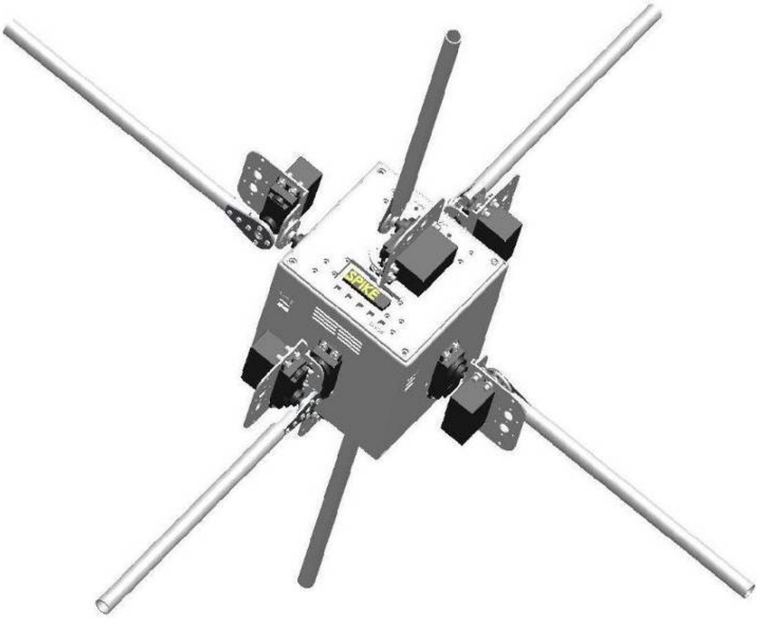


Figure 1-2: Spike CAD Design

The outcome of this thesis will answer this key question: “What is a unique form of robotic motion, capable of being implemented in physical reality on a novel mobile robot?” An additional project attempting to combine evolutionary computation to allow simulated evolution to be implemented in physical reality, by adapting the robotic control was commenced at the same time as this research began. The subsequent withdrawal of the student left this area of development unfinished, and at the time of writing forms a new area for further development.

1.3 Thesis Structure

This thesis is organised as follows:

Section 2 (LITERATURE REVIEW) summarises different forms of land-based robot motion and novel robotic hardware design.

Section 3 (DESIGN METHODOLOGY) describes the research process of creating the motion for Spike, presenting a review of the many ideas considered.

Section 4 (ROBOTIC HARDWARE DESIGN) discusses the hardware systems of the robot, separated into mechanical and electrical subsections. The mechanical subsection reviews the modelling and design of the SolidWorks model used for the robotic cube chassis. The leg and joint mechanism is also discussed in this subsection. The electrical subsection reviews the key aspects of the electrical components used in the design. The wireless technologies used and the control of the robot, separated by a high and low level hierarchy, are also presented.

Section 5 (NAVIGATION SYSTEM) explains the robot’s gait, and operation of the navigation system for the robot.

Section 6 (SOFTWARE) describes key features of the programming of the robot, and the communication structure used to facilitate the robot’s control.

Section 7 (RESULTS) presents the operation of the robot, describing the ability of the robot to follow various headings. Results that indicate the ability of the robot to operate in varying lighting conditions are also presented.

Section 8 (CONCLUSIONS) summarises the topics discussed in the thesis.

Section 9 (FUTURE DEVELOPMENT) discusses recommendations where the growth of the robot’s control and motion may be expanded. As the robot is an excellent platform for development, there are many areas that may be explored in future research.

2 LITERATURE REVIEW

2.1 Introduction

The role of robotics seen in everyday tasks has become more dominant than ever before. Although the definition of a robot includes a wide range of devices with many different levels of autonomy, intelligence and mobility, robots that are able to move effectively in their environment are more important than ever, due to the large number of situations where robots are now used. The origins for applications of robotics are largely seen in industrial situations, where machines have historically been used to replace humans in tedious jobs. This was seen by the first working industrial robot designed in the 1950's, but put to use by General Motors in 1961 [2]. This robotic machine manufactured by Unimation and named the 'Unimate' worked with die-cast moulds and performed welding tasks.

In the last 45 years robotic research has ranged from industrial robots, service robots, underwater robotics, field robotics, construction robotics, humanoid robotics, rehabilitation robots, mobile robotics, research robotics, and more recently, study in biologically inspired robots. With each variation of robot system, a unique method of locomotion is often required. As the research on motion is conducted, there is much emphasis on design and control systems for robotics to therefore be adaptive to their environment, and to be able to deal with obstacles of the real world.

In subsequent years, the complexity of tasks has required robots to have greater flexibility, bringing about research in mobile and walking robots from the late 1960's onwards [3]. From that point, robotics bloomed into a vast number of areas with many different levels of ability, design and autonomy.

In the following literature review, a summary of ground-based gaits used on novel robots is presented, with an emphasis on legged approaches, and including some discussion on robots and robotic motion that is biologically inspired. The legged robotic approach is most relevant to Spike's design, but it is also necessary to examine other forms of locomotion to gain scope for the research. Appropriate review of other forms of robotic motion was necessary to understand any similarities between existing motion, and Spike's motion.

2.2 Types of Robotic Motion

2.2.1 Wheeled Robots

Wheeled robotic usage features in research areas from as far as interplanetary exploration to industrial mining, and has a large number of devices that illustrate their effectiveness. Wheels are generally a preferred mobility system due to their typically fast motion, and ease of control; however when used on irregular terrains such as a soft or slippery surface, it is possible for wheels to become stuck, or to loose traction through lack of friction, and thus become inefficient. Wheeled robotics also faces challenges when confronted with obstacles that are greater than half of the robots wheel diameter. When larger obstacles are encountered, the robot must have appropriate control to navigate and avoid these obstacles.

Although wheeled robotics have advantages over a legged approach, when it comes to stopping on a gradient slope, each wheeled system will typically have a maximum incline level for proper operation, thus very steep hills may become limiting as surface friction decreases. As the friction level of surfaces is important to wheeled robotics, research has been performed on floor-type identification such that their control may become adaptive to the environment. Chang et al adapted a robot to measure the resistance change from brushes placed near the wheels of the mobile robot. With this method, resistance changes in the brush sensors were detected that translated to various surface frictions [4]. The authors were able to use this method to identify friction levels found on three surfaces: wood, carpet and marble, and provided a basis for further investigation of robotic control using this data.

The research of Chang et al is aimed at household robots such as the iRobot Roomba [5], a wheeled vacuum cleaning robot. This robot uses small wheels to move about a household and clean. During the cleaning cycle, the robot must overcome many obstacles by diversion, or else it would become stuck. To achieve this, an anticipation system for predetermining problems was required. This added complexity to the robot control design, but became a balance between operating efficiently in its environment and cost of implementation.

The development of wheeled robotics is not limited to obstacle finding/avoidance, but largely in gait development and deployment. Examples of deployment can be seen in the service industry with the development of meal delivery robots. 'i-Merc' by Carreira et al is presented as a solution to logistical issues faced by healthcare services for the "transportation of meals from the kitchen to the patients' rooms and for returning the respective soiled dishes safely, preventing all possible contamination, be it through the meal, or the service personnel" [6]. A motivating factor behind their research was largely based on studies performed by the hospitals and health care centres indicating the drawbacks of traditional delivery methods, including issues surrounding weight and handling. The use of wheeled robotics in this application was effective due to hospitals' usual tendency to have hard, smooth surfaces with limited variations of change in surface consistencies. By using a wheeled robot in this application, the robot was able to move in a smooth controlled manner suitable for the stable and effective delivery of meals.

Wheeled robots are also combined with other similar mobile robots to form a branch of robotics known as modular robotics, discussed later in this chapter. In [7], Gaston et al reviewed the effectiveness of a new type of mobile robot that was based on wheeled locomotion, and attempted to overcome the difficulty of a small robot's ability to traverse common obstacles. This application allows some traditional limitations of wheeled robotics to be overcome by the use of the 'Alliance' architecture [8], and modular robotics. The Alliance architecture allows behaviour based on cooperative actions between robots to achieve fault tolerance by adaptive action selection. The authors developed three modular robots that use a scissor hoist type action for lifting each of the robot modules up a flight of stairs, with wheels for forward propulsion and distributed programming to achieve stair climbing ability.

Gait development of wheeled robots is also possible with standardized 'off the shelf' robots such as the Khepera II platform [9]. The Swiss made mini wheeled robot manufactured by K-Team is used around the world as a robotics development and testing platform. It has become a standardized robot platform for research on wheeled robotics in over 500 universities and provides numerous sensory inputs and control outputs for development. The cylindrically shaped robot measures 55mm diameter with 30mm of height [9]. There are two small direct current (DC) motors that drive the robot, and an extension data bus to allow communication with custom external devices.

Another use for a wheeled robot in research has been mobile inverted pendulums (MIP) [10, 11]. Perhaps one of the most recognised MIP's commercially available is the Segway, a mobile transportation platform with two wheels. In this system, the robot's balance is controlled by two wheels, such that ‘standing up’ becomes possible for the robot. The two wheels are usually combined with gyroscopes and other feedback mechanisms to allow the robot to maintain its inverted pendulum stance.

2.2.2 Legged Robots

Legged robots have many advantages over wheeled robots. Legged robots are able to move effectively, with only a few legs in contact with the ground surface. Wheeled robots in comparison move most effectively when all wheels are in contact with the ground surface [12]. For this reason, legged robots are more suitable where the ground surface consists of natural, uneven terrain, typically seen in outdoor environments. Legged robots are able to step over objects [13] and traverse up and down slopes [14], as well as vertical walls [15] and even move on soft surfaces such as sand, mud or snow [13], where wheeled robots may become stuck.

The shortcomings of legged robotics are associated with the increased technical complexity of their mechanical and electrical design. Each limb of a legged robot requires several linkages to facilitate motion, and thus complex electronic systems must be designed around this. This induces an effect on the total price of a legged robotic system, and typically means legged robots are a more expensive option.

Robot stability is a key research topic for legged robotics, where most legged robots are defined as stable when they are able to maintain their balance without external assistance. The interest in walking robot stability first came from the research of McGee and Frank in (1968), in which they first defined the static stability of an ideal walking robot [16]. Static stability proved to be limited, as it did not consider the effects of inertia experienced by the robot during motion by limbs, body, friction, and thus limited the robotic movement. Subsequently, research on dynamic gaits arose that allowed the development of gaits that could trot, canter and gallop on a quadruped robot [3].

To analyse legged robots, a practical way is to form categories based on the number of legs used by the robot. Legged robots may have one [17], two [18], three [19], four [20], five [21], six [22] or even eight legs [23] or more. Because there are so many variations of legged robots and hence a large number of gaits possible, the following chapters will focus on a selection of interesting novel gaits for land-based legged motion.

2.2.2.1 Bipedal Robots

Bipedal robots are commonly known for mimicking the action of a human walking gait, but are not always in the form of a humanoid. Bipedal means the robot has only two legs that are used in conjunction with some control to be stable, and move. There has been a great deal of research performed on bipedal robots [3, 18, 24-27]. As bipedal robots are inherently unbalanced, there is a heavy requirement for the development of efficient control methods to maintain stability in these designs, while allowing locomotion. Although there are many variations for achieving bipedal stability and locomotion, two commonly used methods are: control of the zero moment point (ZMP) [28], and the principle of dynamic walking stability [29]. The ZMP is used on ASIMO by Honda [18] and WABIAN-2 by Waseda University [24], and examples of the use of dynamic stability control are the Cornell Efficient Biped [30] and Denise from Delft University [31].

The Zero Moment Point (ZMP) concept was introduced by Vukobratović in 1968, and provided a point on the floor (the supporting plane) that was determined by the gravitational force and inertial forces of a walking gait [12]. Graphically, this point is normally seen as the point on the ground directly beneath the robot’s torso. By controlling this point, the robot may take on forward motion while remaining dynamically balanced [32].

By using the dynamics of the robot, stability may be achieved as McGeer proved with his totally passively dynamic stability robot that was able to walk with stability down a small incline [29]. This periodic motion of the legs was proven to be very efficient as it required minimal control, and in some cases only minimal input energy from an external force (motor). Rather than using a fixed point to perform calculations and derive the robot’s locomotion from as with ZMP, dynamic walking uses a pendulum swing action that bipedal robots exhibit inherently. Through accurate mechanical planning and control, the motion of each leg’s swing action combined with correct foot curvature, enabled the natural walking to occur. McGeer proved that with accurate and planned mechanical design, the leg motions led to a natural walking motion and was highly efficient [33].

Current development of bipedal locomotion includes improving motion to better match a human walking gait, and being able to tolerate obstacles. Hoonsuwan et al analysed a human walking gait for replication on a bipedal robot [34] while using ZMP for stability. The authors successfully implemented a human walking gait by replication of fixed points attached to a live human leg in walking motion. Three fixed points for each human leg were fed into a computer program using image processing techniques that enabled data extraction from the human space to the robot space. This data was then used to form the gait for the robot's locomotion.

Takubo et al investigate a simple 'step-up' and 'step-down' system for traversing over small obstacles with a humanoid robot [35]. This system provided real-time modification of ZMP criteria to allow the robot to handle stepping on and off a multilevel surface. By investigation, simulation and testing the HRP-2 robot [36], the authors boasted a successful gait for stepping from one surface height to another. Although the 'obstacle' presented was a minor example of a real life obstacle, the concept is extremely relevant for growth into real world situations.

2.2.2.2 Tripedal

There are very few robots being used in research that feature only three legs. This may be due to there being no three legged animals in our human environment. As there are no three legged insects or mammals, robotic designs in this field cannot be mimicked from biological forms like other biped or quadruped robots. A brief tripedal stance may however be observed by Cockroaches during their standard swing phase [37], and a tripedal stance may be seen by resting Kangaroos using their two legs plus hind tail to create a tripedal platform. Other multi-legged robots may take a brief tripedal stance during their locomotion [38, 39].

In 1988 a proposal for a three legged robot for lunar base construction was drafted [40]. One of the benefits outlined by the publication was the advantage of static stability using three points of contact with the ground. Since then, recent examples of tri-pedal robots have used the benefits of inherent stability, with different gaits to form a unique locomotion. Two recent developments are: Rotopod, a rotating legged robot developed by the Robotics and Computer Vision Laboratory of Fordham University [41], and STriDER, developed by the Robotics and Mechanisms Laboratory of Virginia Tech [42].

Rotopod's gait was created by rotating a mass extended from the robot chassis around the robot, to allow the energy from the rotating mass to be transferred to the legs. The mechanism operated in a similar method to a rotating wheel, with the robot turning around a single point (leg), as the mass rotated around the chassis. The robot was able to control its travel direction by changing the length of each of its legs, such that the angular momentum of the mass caused the robot to spin around the shortened leg. The gait created was therefore of a spinning forward motion.

The STriDER robot used three legs with dynamic walking for its locomotive propulsion. The locomotion was generated by the robot alternating between stable and unstable standing positions, using a single leg to shift and swing beneath the robot toward the bi-sector pivot line of the remaining two legs. This ability was assisted with an actuated hip, pelvis and knee joints to create the necessary swing. With this action, motion in the desired direction of travel was possible, with changes in direction implemented by alternating the sequence of leg swings.

2.2.2.3 Four and Six Legged Robots

There are many robot designs that feature four or more legs. Many multi-legged robots attempt to tackle the challenge of locomotion on uneven terrain, in environments that are unknown [38] by using robot navigation [43]. With an increased number of legs, various forms of gaits are possible such as a rolling gait [44], that would be typically be likened to a spherical chassis.

Pai et al have conducted research and have developed a new family of multi-limbed robots known as Platonic Beasts [45]. The robot presented by the authors in this publication features an octahedron shaped chassis, with four leg limbs each with three degrees of freedom, that were used to create a rolling gait. The limbs rotated the chassis above the ground around a central axis using a series of movements and steps that allowed new leg configurations to be formed and motion to occur. The robot’s unique design created no inherent orientation to the ground surface and also allowed motion by rolling and tumbling, which provided a better recovery rate from obstacles and uneven surfaces. This feature was useful should locomotion on rough terrain be desired.

Boston Dynamics has conducted a large amount of research on robotic gaits. Their engineers have conducted investigations over 25 years that have led to the development of hopping, running, trotting, pacing, climbing and jumping gaits, and have created robots that have set land speed records. Their BigDog robot [13] has been constructed as a quadruped robot capable of locomotion over real world rough terrain. Their cumulative investigation of different gaits facilitates the ability to manage natural terrain changes in the real world environment, such that their successful design is able to travel 2.2m/s when running and can carry around 154kg. The BigDog adapts to the terrain by adjusting its body height and its footfall placement to compensate for the robot's body orientation relative to the ground plane. Other researchers have investigated control architectures for LittleDog, the 'little brother' of the BigDog robot [20, 46].

Another multi-legged approach in robotics uses six legs. Such hexapedal robots often take biological inspiration, as with quadruped and bipedal robots. Feilding et al created a hexapod walker named HAMLET [22] that used force and position control to achieve an adaptable walking gait on rough and slippery terrain. The authors developed a 'soft' control architecture to derive the gait control. This architecture facilitated the ability to modify the robot parameters in real time, using a PC serial link. The components of HAMLET are given names like 'Coxa', 'Femur', 'Tibia' and 'Tarsus' for simple identification of their position and usage. This robot resembles insects such as the Carausius morosus ('stick insect').

Hexapedal robots do not always use motors or pneumatics for direct leg manipulation. The RoACH autonomous crawling robot [47] by Hoover et al, is a miniature sized robot that uses smart composite microstructures (SCM) in conjunction with shape memory alloy (SMA) wire for locomotion. The authors are able to articulate the legs of the robot from the force of contracting the SMA by using pulsing from a microcontroller. The robot weighs only 2.4g and at the time of paper publication (2008) is claimed to be the smallest and lightest autonomous legged robot produced.

2.2.3 Climbing Robots

Another form of robotic motion is a climbing gait. Climbing robots may be used to overcome obstacles like stairs when used in conjunction with other modular robotics [7], however another interesting application of climbing is seen with autonomous traversal up pipes [48] or flat surfaces [49]. The hexapedal climbing robot designed by Grieco et al is designed with transporting payloads up to 100kg vertically up flat walls and on ceilings using ferromagnetic surfaces [49]. Each leg uses electromagnetic contacts as feet to maintain the robot's position on the surface.

Another variation of a legged climber is RiSE V3, a rapid pole climbing robot [15] developed by Haynes et al at the University of Pennsylvania and Boston Dynamics. This robot utilises a dactyl claw system as its foot mounting apparatus, with platform verification performed by climbing wooden objects such as a telephone pole. The claw climbing system is effective due to the mounting angles of the claws, to allow its gait to repeat at a rate of 1.75Hz, while maintaining traction during climbing. The authors discuss future work to include a faster gait for the robot and a recovery system for coping with a slipped foot condition.

The Alicia II climbing robot for industrial inspection by Longo et al, was developed with a gait that allowed vertical wall inspection of petrochemical tanks, as well as building inspection [50]. The authors identify hazards associated with forms of inspection, and justify their design as a safety robot that may be used where humans may find access to inspection areas challenging or dangerous without the aid of expensive equipment. The Alicia II propelled itself by using wheels, and maintains vertical surface pressure via a vacuum suction system. This suction was created with motors that ran continuously during operation, to keep the robot's cup shaped chassis at a negative pressure to the wall. The clearance between the robot's cup rim and the wall was lined with brushes and material to assist with maintaining the vacuum pressure. This work has been extended to a simulation of a third generation of Alicia where three of the second generation modules were linked as a chain. This provided the ability to overcome obstacles by detaching a single unit from the wall, while still being attached with the remaining two modules. The detached module may then be repositioned to overcome the obstacle.

2.2.4 Hybrid & Other Robots

2.2.4.1 Snake Robots

Oscillation motions are proven as an effective gait for locomotion [51]. Research on oscillating gaits has seen robots take biological inspiration from animals such as snakes [52]. Hirose played an integral part in the development of snake like motion [53]. Examples of snake gaits include: sidewinding, swimming, lateral undulation, self-excited, linear progression, cornering, pipe rolling, channel climbing, rolling and many others. Their motion enables navigation to overcome obstacles, and has the ability to move effectively through complex environments [54, 55]. This motion was assisted by the many small linkages that make up the chassis of the robot, allowing hyper-redundancy [56, 57].

The snake-like chassis may also be used with tracks to achieve locomotion. This was done by Junyao et al with their robot designed for rescue applications [58]. The robot featured many small tracks mounted around the face of each segment of its body that allowed the benefits of wheeled motion for the snake, such that it was able to pass through small gaps that may not have been possible with a regular serpenoid curve or other snake gaits.

2.2.4.2 Hopping Robots

Another form of robotic motion is taken from a hopping gait exhibited biologically by a dog's leg [17]. In this publication, Hyon et al simulated the use of muscles and tendons by utilizing two hydraulic actuators and a spring on a single leg. A significant contribution in hopping motion was contributed by Raibert [59] by developing the first self-balancing hopping robot, which has been extended to provide research openings on new hopping robots used for planetary exploration [60], and may include a higher number of hopping segments [61].

2.2.4.3 Modular Robot Motion

'Modular robots' is a vast area that includes many levels of design, gait, and autonomy [62]. Modular robots are small robotic units that are connected together to perform operations as a group, that would typically require a specialised system. The mechanical design of modular robots can be broken into two areas of design: homogeneous and heterogeneous systems. A homogeneous system is one where all modules are of identical form to each other [63], and heterogeneous systems include two or more different modules [64].

Both homogeneous and heterogeneous modular robots are seen typically as small units, and have the ability to achieve a task by reconfiguring or deforming their overall structure by modifying their interconnections and hence physical form [65, 66]. Modular robotics allows the self-reconfiguration of the physical robot's form, thus providing a variety of gaits [67]. Modular robots may configure themselves to move as a legged motion, wheel type motion, oscillating motion and many other non-standard forms [66, 68, 69]. Many of the robots functions may be first simulated on a PC before being tested in physical reality [70].

A well developed example of homogeneous modular robotics with many levels of robotic locomotion, is the M-TRAN series of robots developed by the Advanced Industrial Science and Technology (AIST) Institute of Japan. AIST has over 35 publications from their M-TRAN robots [71], including a US Patent. Now in their third generation of development [72], the modular robots are able to reconfigure into countless different shapes to create different patterns of movement. Each of the modules feature their own processing abilities, power supply and ability to sense and manipulate their connection to other modules. The physical construction of a module takes the form of two cube shaped blocks linked such that the module's control may be separated into a passive and active block. When the active and passive blocks of a module are linked to other module's blocks, together they are able to perform an increased number of operations with their increased degrees of freedom and physical form.

'Odin' is a novel heterogeneous modular robot design whose motion is derived from 8 telescopic links developed by Lyder et al [66]. The deformable lattice design is novel due to its ability to extend itself via its telescopic modular linkages. The robot also features special joints that provide connectivity sites for adding new modules to perform specific tasks. The ability to add new and different modules to the robot provides a platform for building up a sophisticated robot allowing the robot to be best suited for its environment. Lyder et al note that their design is able to adapt to the environment by deforming, rather than adapting by changing the robot's configuration. As the design can be built upon with new parts, the motion of the robot may be better suited for the environment. The actuation of the linkages on the robot creates a wriggling type motion as the lattice structure deforms for its gait.

2.2.4.4 Combined Leg and Wheel

Another approach to robot motion has been to attempt to combine the benefits of fast moving wheels, with rugged terrain handling legs. Such 'leg-wheel' robots are able to move their combined leg and wheel drive systems to access or move on terrain that would normally be accessible to robots that move with only a single version of the drive system [73, 74].

There are many robots that use a combined leg/wheel design [75-79]. A popular approach utilizes a WHEGS design. WHEGS is leg-wheel design that uses the principles of a wheel type system to rotate a number of spoke-like legs, which in turn create a circular wheel type motion for the robot. WHEGS designs may also be altered to include adhesive tape as feet for vertical wall climbing [78]. Typical applications of WHEGS designs include four and six leg/wheel robots, and have been documented to overcome obstacles such as stairs and uneven terrain [76]. Other designs in research have included usage of wheel/track/leg as their drivers for locomotion such as MOBIT [77], and wheels as the feet of hexapedal robots such as ATHLETE [80, 81].

2.2.4.5 Spherical

Robots that use a spherical shape for their chassis are able to generate a rolling motion for their gait, due to the similarities between spheres and wheels. The rolling motion of spheres differs from wheels by their nonholonomic ability. Examples of spherical robots in recent publications utilise forms of an internal weight shift mechanism within the sphere, to create an imbalance of the centre of mass [82] and also use principles of conservation of angular momentum [83] to achieve locomotion.

Spherical robot design allows the robots to roll over small obstacles, and move with a degree of agility. This can be aided with the use of a camera [84, 85] for navigation or via manual control [86]. It is also important for robots of this type to be robust enough to handle unstructured environments as damage to the chassis will directly affect the locomotion. Planning and consideration of such factors is discussed in the construction of ‘RoBall’ [87].

3 DESIGN METHODOLOGY

3.1 Overview

This chapter presents an overview of the concepts investigated for creating Spike's motion, and includes discussion and analysis with illustrations of the chosen method. The chapter outlines the scope of the entire project and identifies how the robot is operated. It will expand on the research question by introducing concepts investigated for creating the unique form of locomotion for the novel robot platform, and concluding with a description of the actual mechanism used in the design. The evaluation of the mechanism is enhanced with the development of a navigation system, while test results obtained by using the navigation system are discussed in later chapters of this thesis.

The research in this thesis is centred on the discovery of a novel way to generate robotic motion that has not been explicitly explored before. To facilitate this, the development of specialised hardware was required. The development and construction of the robotic hardware was therefore a major component of creating the novel robotic motion, and allowed the research question to be fulfilled. The development of the robot is discussed in chapters 4, 5 and 6 of this thesis.

The robot's locomotion was based around a cube chassis form, and constructed with a single leg protruding from each face of the cube. The robot used three axes of symmetry with a collinear pair of legs mounted on each axis. This provided six legs that the robot was able to manipulate. The mounting of the legs on the chassis created a tripedal stance when the robot was standing in a static position and hence was inherently stable. As the robot was inherently stable and holonomic, a method of creating motion for the robot was determined by forcing a change in the robot's centre of mass by reconfiguration of the leg position in the tri-pedal stance.

Many concepts were considered, based on an idea of shifting weight to allow Spike to move. Each of the individual concepts was assessed for its motion potential and effectiveness. Many of the methods investigated dealt with a series of weight shifts by the legs to allow the robot to move. It was discovered that nearly all of the concepts exhibited a single problem; although changing the dynamics of a single leg was possible, all legs of the robot system would have the same properties and hence, the robot would remain in a state of inherent balance and stability. Although there were many methods of creating motion for the robot, the chosen design required a balance of simplicity in its design and a cost effective use of parts.

In the following sections, concepts are presented that show the research process of creating motion for the robot. Emphasis is placed on the creation of a tumbling motion for the robot as its form of locomotion. The chosen design for creating tumbling motion is presented with details of its use to facilitate a type of gait. The design of the robot framework is also presented where the envisaged motion concept may be applied. A basic model of the robotic frame is detailed with a set of possibilities that each leg may make in order to achieve its objective of motion.

3.2 Ideas & Concepts

3.2.1 Basic Model

The framework for Spike's design began with creating numerous concepts from plastic parts that formed both interesting and creative shapes. This facility was provided with the use of several small linkages and interconnects, joined like building blocks to construct creative chassis designs. The purpose of using many interchangeable linkages was to quickly design a novel shape that exhibited a unique design, with a non-standard capacity for motion. Although many chassis concepts were constructed, it became apparent that many of the designs built with the plastic parts had been previously developed and published. This discovery was made in conjunction with the literature review process.

After the analysis of several motion concepts shown in this chapter, a novel shape was constructed using a design that resembled the 'Czech hedgehog' military anti-tank obstacle. The shape had a central chassis with protruding limbs that had the ability to be used as legs. The basic model of this design is shown in Figure 3-1. From the literature review process, it was determined that there were no robotic designs that resembled this shape, which meant the subsequent motion required to move the shape would therefore be a unique method.

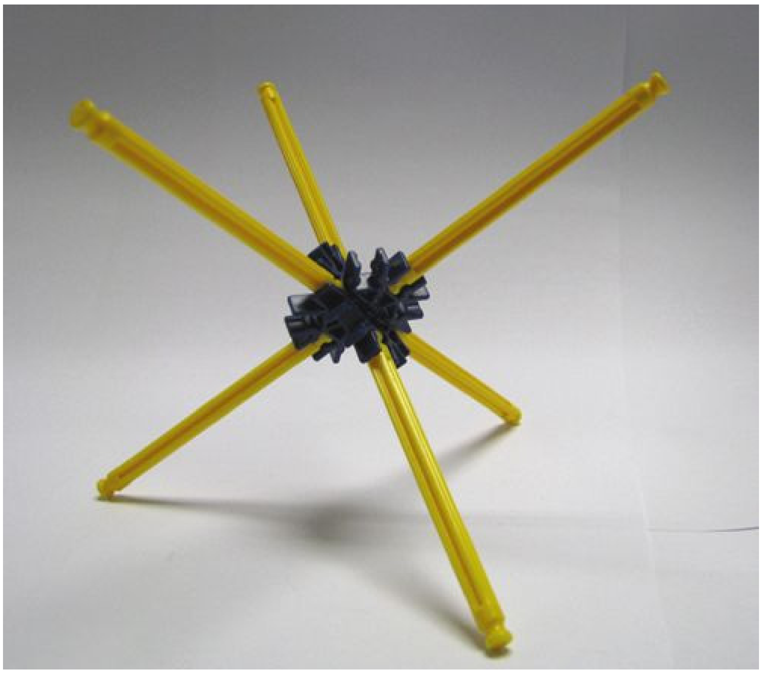


Figure 3-1: The Basic Model

This basic model used three axes of symmetry and had inherent stability with its tripedal stance. The design used a pair of legs mounted on each collinear axis, providing a total of six legs. The leg configuration created a total of 8 possible 'surfaces', where a 'surface' was a region defined by three feet that created a triangular footprint when placed simultaneously on the ground surface. The combined surfaces formed by the feet of the legs created an octahedron shape, which was used as a reference for describing the orientation of the robot. The design had no inherent orientation in its environment to indicate whether it was 'up' or 'down', and thus required a method of interpreting its orientation to become controllable regardless of its orientation. The inherent stability posed by the robot's footprint is advantageous when the surrounding environment is uneven or unstable; however the design requires some manipulation to force instability if motion is required.

In the basic model, this instability was caused by actuating the legs in such a way, to create a tumbling and falling action by shifting the robot's centre of mass between stable and unstable states. The following section discusses the concepts that were considered for creating Spike's motion as part of the research process, and outlines the final concept selected for the robotic prototype.

3.2.2 Shifting Weight

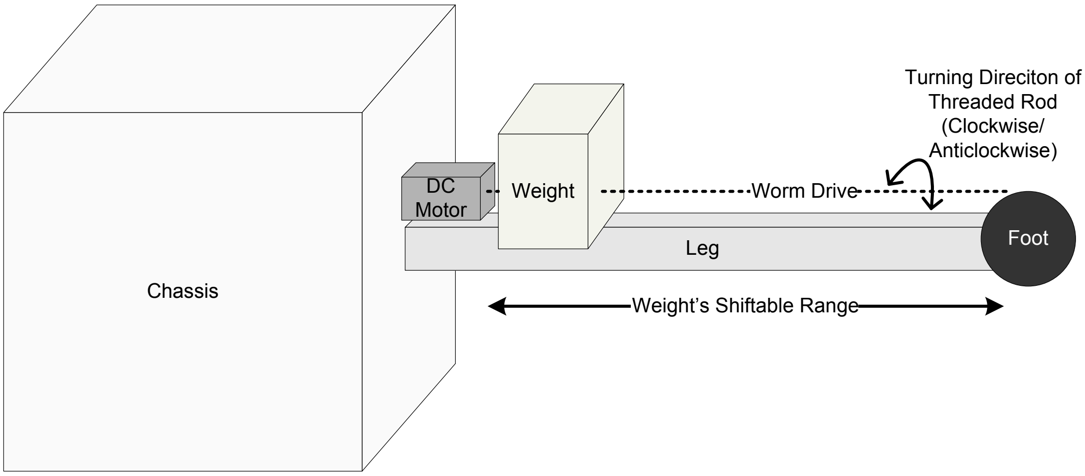


Figure 3-2: Weight Shifting Concept

The concept illustrated in Figure 3-2 shows a shifting weight mechanism. The principle of this idea for generating motion for the robot was based on manipulating the robot's centre of mass, by a sequence of weight shifts generated from the leg mechanisms. The leg mechanism would operate by using a custom moulded weight that includes a threaded hole through its centre, to allow a threaded rod to be inserted. The weight may be moulded to be of rectangular form and sit on top of the leg itself, or of a circular shape to encompass the leg. The threaded rod would be rotated by a DC motor, mounted at a single end of each leg. By rotating the threaded rod using the DC motor in either a clockwise or anti-clockwise direction, the weight would slide along the threaded rod, next to the leg, until the DC motor stops rotating the rod. The process of shifting the weight on each leg extension induces a change in the robot's centre of mass. It was considered that with an appropriate sequence of weight shifts by all six legs, the centre of the robot's mass would shift from a stable balanced state to an unstable state, and therefore tumble between leg configurations toward the desired direction of travel. Two of the three legs on the ground would act as a turning fulcrum, and the main weight shifting leg facing the direction of travel would instigate the motion. Although the idea is conceptually plausible, the inherent stability of the robot's tripedal stance poses problems for this design. It was determined through analysis using plastic models, that the likelihood of the centre of mass to shift substantially past the turning fulcrum was small, as each leg used the same weight mechanism, and thus was assumed to always be a stable system. This can be explained from the turning moment created from extending the weight along the leg, on the falling side of the cube, which would be opposed by forces of the opposite legs, and also by the adjacent legs of the robot. These forces from the other legs would act against the moment being created by the primary weight shifting leg. These factors deterred the use of this system. Further analysis may conclude a derivative of this design to be successful; however for the development in this project, a new design was sought. The application of the shifting weight leg mechanism on the basic model is shown in Figure 3-3.

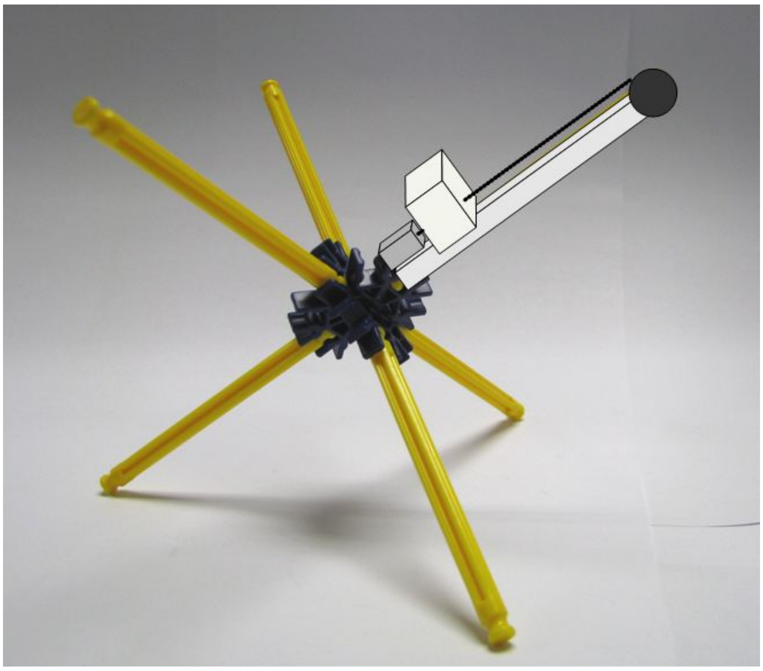


Figure 3-3: Shifting Weight Concept on Model

3.2.3 Water/Air Jet

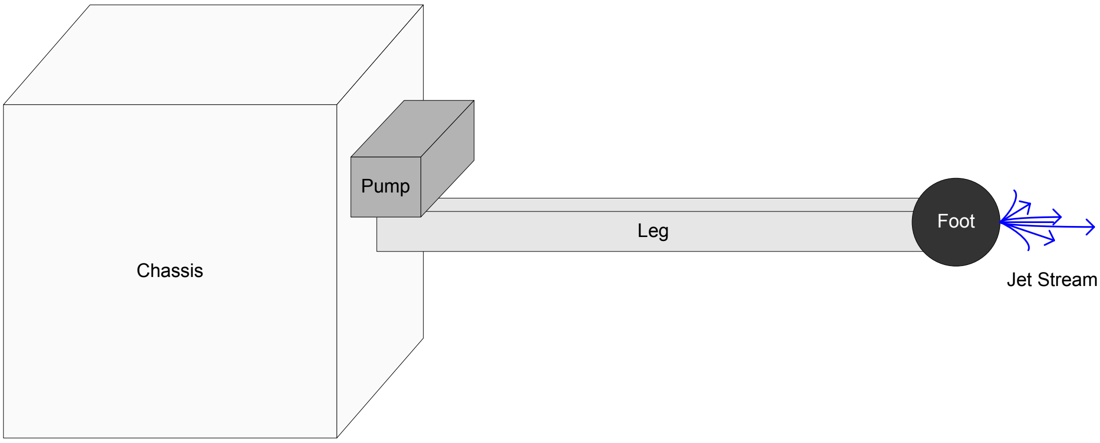


Figure 3-4: Jet Concept

Air and water are two very powerful natural forces when used in a controlled way. A novel concept for creating Spike's motion could have been be to use one of these natural forces to propel each leg from the ground to cause tipping. This would be achieved by forcing compressed air or water stored from within the chassis, through the leg and to shoot from an orifice on the foot to propel the leg away from the ground surface. This mechanism could be further developed by using a bladder system to exchange water between the centre cube chassis and a bladder located at the extremity of each leg. By using a bladder system with a bi-directional pump, the water displacement would theoretically act as a ballast, to create a turning moment for the robot. By controlling this turning moment, the tumbling motion for the robot might have been achieved. After further consideration, it was decided that as water posed hazards to the electrical systems of the robot and surrounding equipment, the water concept should be replaced with a concept of air.

The operation of a compressed air system within the cube chassis would require a method of either generating compressed air by means of an internal air compressor, or the ability to interchange rechargeable canisters holding compressed air. The air would be blasted from the orifice of the foot at such a rate, that it could overcome the sum of the forces acting to keep the leg on the ground, with enough power to blast the robot forward into a new leg configuration. The air pressure stored within the robot's compressor system or rechargeable canisters would be stored under significant pressure to hold enough energy to enable successive tumbling steps to be made. Initial concerns for implementing this design were based on power-to-weight ratios for the robot systems, and the quantity of force required for the air jet stream to function effectively. Compressors and engine systems are typically heavy devices, and would add to the overall weight of the robot if included within the robot's chassis. Additional weight would need to be offset by the propulsion system, and hence to cause enough propulsion from an air jet would require a significantly powerful engine system, and would in turn, increase the robot's weight. It was deemed that this design would be too mechanically challenging to achieve in the timeframe of the project.

The air propulsion concept may also be only effective in environments where the ground surface is firm and consistent. For example, if the air jet were to be used on a soft, sandy surface, the air jet may cause a large disruption to the surface when activated and begin a process of creating a hole that the leg would sink into, and therefore have little or no forward motion effect and a greater opposing turning moment. An example of the jet stream concept is illustrated on the basic model in Figure 3-5.

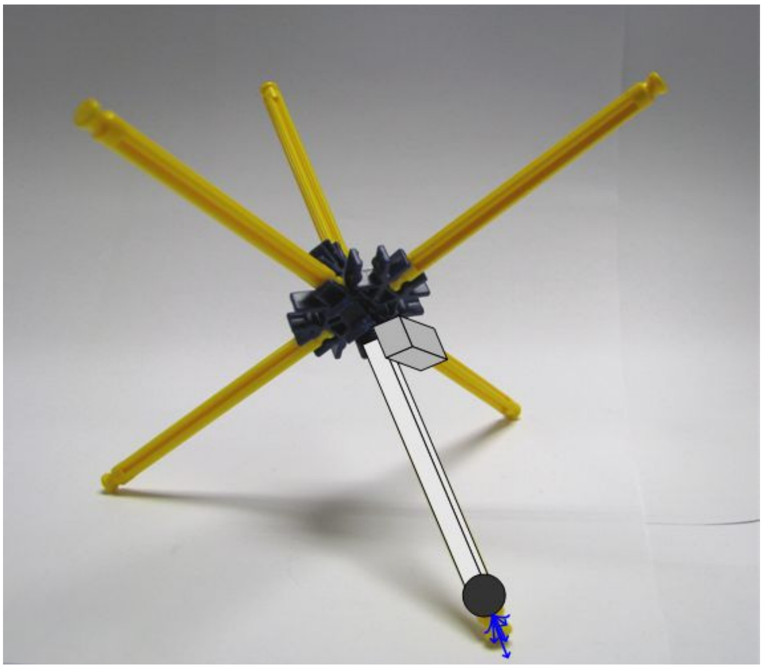


Figure 3-5: Jet Concept on Model

3.2.4 Catapult System

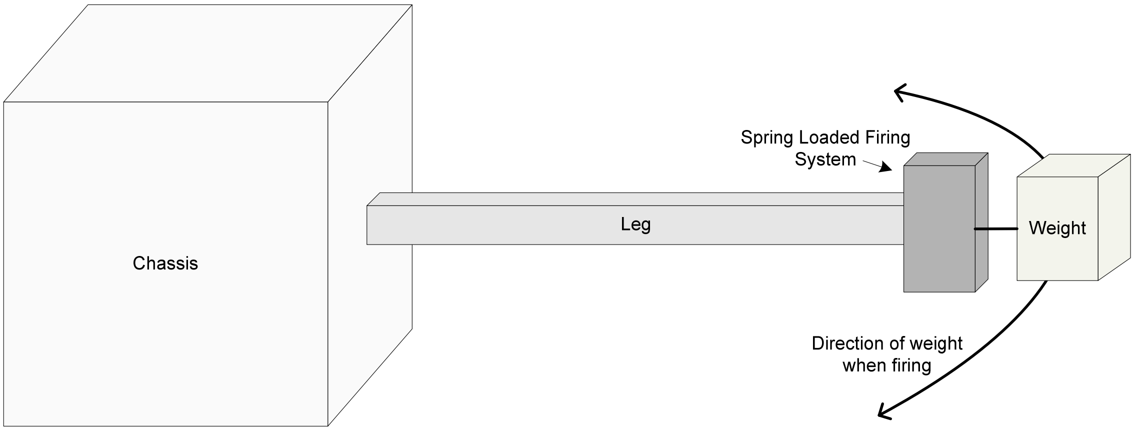


Figure 3-6: Catapult Firing Concept

The purpose of a catapult is to hurl an object through a range of motion. The catapult motion may have been applied to the Spike robot to induce a forward momentum, which would have allowed the robot to swing forward into new leg configurations. This system would be generated by the use of a spring loaded firing mechanism to hurl the weight in the direction of travel. The momentum of the weight would have a requirement to be sufficient enough to allow the robot to tip forward in a controlled way. The firing mechanism could be charged by the use of a specialised DC motor, driven from the electrical systems of the robot. The attachment between the firing mechanism and the weight would consist of a flexible, strong material that would allow consecutive tumbling steps to be achieved without being detrimental to the entire propulsion system and robot.

It was decided that the mechanical challenges of implementing this design would increase development time significantly, such that an alternative method was sought. There were also many potential problems posed by this design which would need to be carefully considered. The impulse effect from the motor may become detrimental to the robot, as each leg would act as a lever against the chassis upon activation of the firing mechanism. The momentum required to overcome the robot's stable position would also need to be substantial to overcome the effects of the opposing legs and their weights. For these reasons, it may become difficult to reliably control the motion of the robot in this manner. The conceptual view of this mechanism is illustrated on the basic model in Figure 3-7.

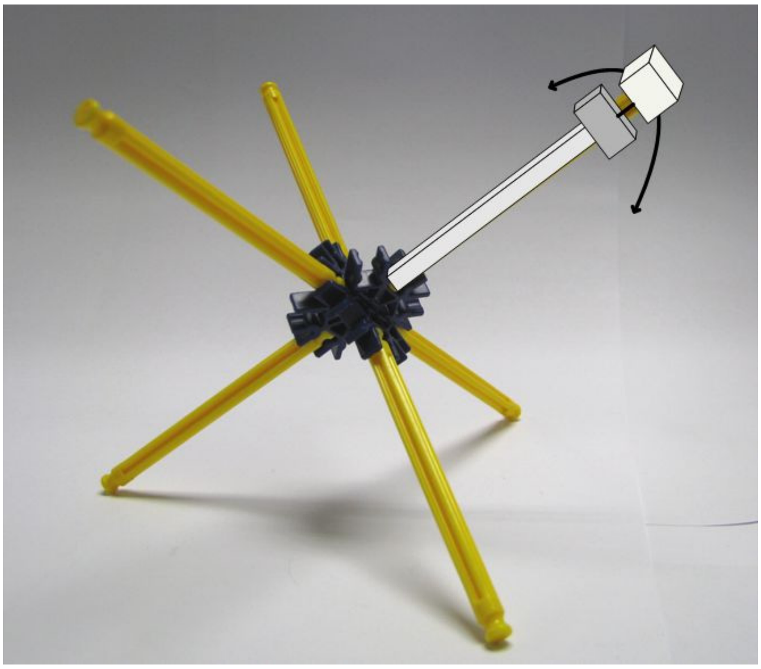


Figure 3-7: Catapult Concept on Model

3.2.5 Telescopic Leg

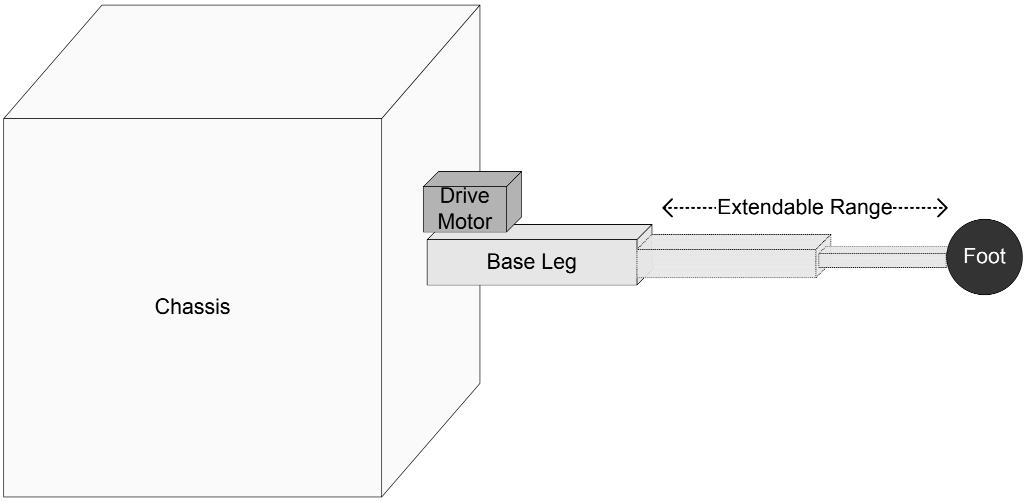


Figure 3-8: Telescopic Leg

An interesting method for generating Spike's motion was considered with use of telescopic extensions. This method would require each of the robot's legs to be telescopically driven, meaning the locomotion would be generated using a series of extensions and compressions of the telescopic legs to create a tumbling motion for the robot. It was considered that shifting the robot's centre of mass would be achieved by shortening two of the standing legs, while extending a third leg against the ground surface. The 'catching' leg, (i.e. the leg directly opposing the extending leg) could also be extended to generate a larger turning moment about the two shortened standing legs.

It was determined that the weight of the centre chassis would pose a large effect on the system's balance, and would be considerably difficult to operate where the ground surface gradient was not consistently flat. The telescopic mechanism would require a high extension ratio while being physically small in its compressed form, to not become cumbersome to the robot design. The telescopic extension would need to be robust enough to tolerate impact from the robot's tipping and falling action, and to support the weight of the robot system. This method was assessed as a more favourable than the other methods described, due to its simplistic mechanism. The design is disadvantaged where a surface is of some raised gradient. In a situation where the robot is attempting to climb a gradient, the tipping action would be more challenging to achieve, as the turning moment that would be required to overcome the effects of the incline, in addition to the robot's weight and angle to the ground, would be substantially increased.

The concept also presented challenges from its own chassis weight. The chassis weight would induce a considerable affect on the ability for the robot to tip and fall between leg configurations. Any weight present at the feet would need to be considerable to overcome the effects of the opposing turning moments, and the opposing forces presented by the chassis. An illustration of the telescopic concept is illustrated in Figure 3-9.

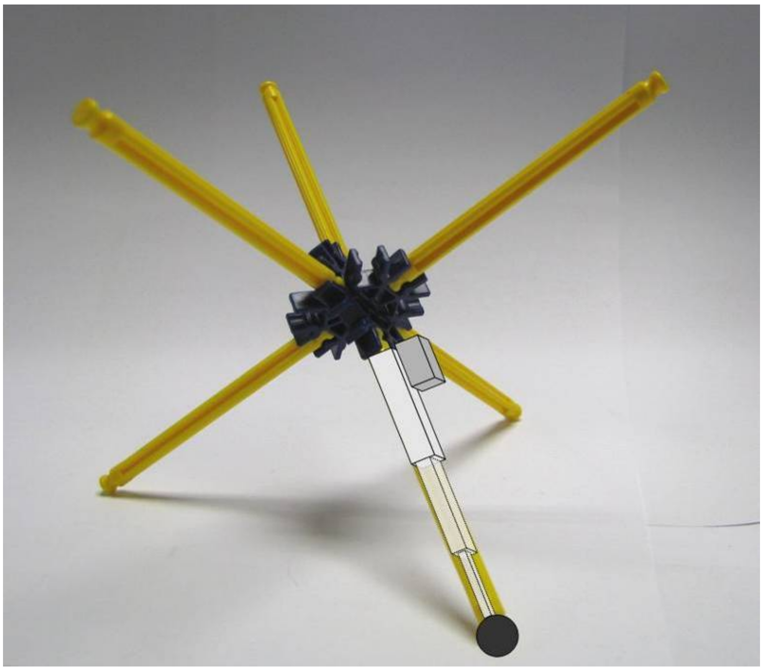


Figure 3-9: Telescopic Concept on Model

3.3 The Proposed System

3.3.1 Overview

The static configuration of the robot forms an inherently balanced tripedal stance, and the legs are manipulated by the control systems of the robot to force the robot to tumble between leg configurations as its locomotion. The legged robot design uses a total of six legs that are each fitted to a single face of its cube chassis. The tripedal stance is formed by three legs contacting the ground surface and three legs facing upward. Light sensors are attached to each foot, which allow the robot to determine which three of the six legs are in contact with the ground to allow the robot to determine its orientation in its environment.

The tumbling and falling motion of the robot was generated by a series of leg actuations that enabled the robot's centre of mass to be shifted from a balanced state, toward the direction of travel. This was performed by the use of a joint mechanism at the base of each leg, powered by two titanium geared servo motors. Each motor was capable of travel through 180°, thus the foot of each leg was able to be positioned at any point around a half sphere from each face of the chassis. This flexibility allowed the leg to be driven to cause motion, regardless of the robot's orientation, providing necessary freedom for the robot and for motion to occur.

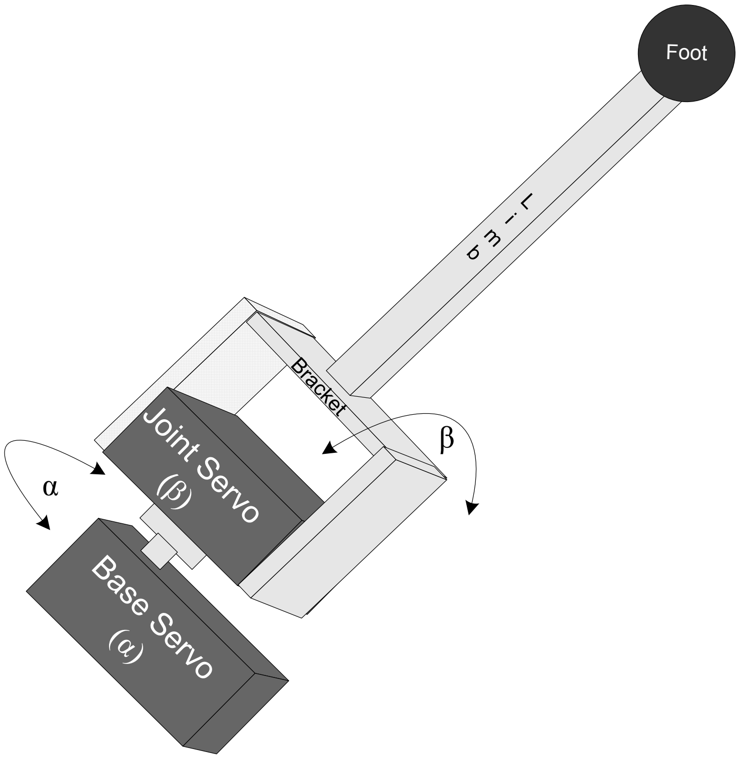


Figure 3-10: The Servo Motor Joint

3.3.2 Leg Mechanism

Figure 3-10 shows the proposed drive mechanism for the robot, Spike. Each leg uses its own pair of motors to individually articulate the leg, such that the leg configurations of the robot were able to be synchronously manipulated to form a gait. The joint mechanism used standard off-the-shelf brackets and parts combined with a custom cube design to create a unique robotic motion and framework.

The joint mechanism operated by controlling each of the servo motors simultaneously, such that each leg was able to be positioned and driven against the ground surface toward the centre of the robot, causing the robot to tumble. The joint mechanism in Figure 3-10 provides motion through two degrees of freedom for each leg. Both α and β rotations provide a range of motion that positions the foot of each leg at any point around a half sphere, created from the face of the cube by each joint mechanism.

The first ‘base’ motor (α) rotated the second ‘joint’ motor (β) that articulated the leg, such that the foot was driven toward the midpoint of the opposing triangular side and hence, the robot fell in line with the driving leg’s direction. For Spike’s gait, the base motor’s rotation was limited such that the foot’s direction of travel would be in line with the ground, and travel toward the midpoint of the opposing side. This was achieved by limiting the base motor’s rotation to ±45° from its start position, and tilting the joint motor by a full ±90° from its starting position.

The combined servo motors’ ranges of motion when mounted in the joint mechanism, provided the ability for each of the feet to be driven in one of the four required directions, determined by the programmed sequence of movements in software. This is illustrated in Figure 3-11.

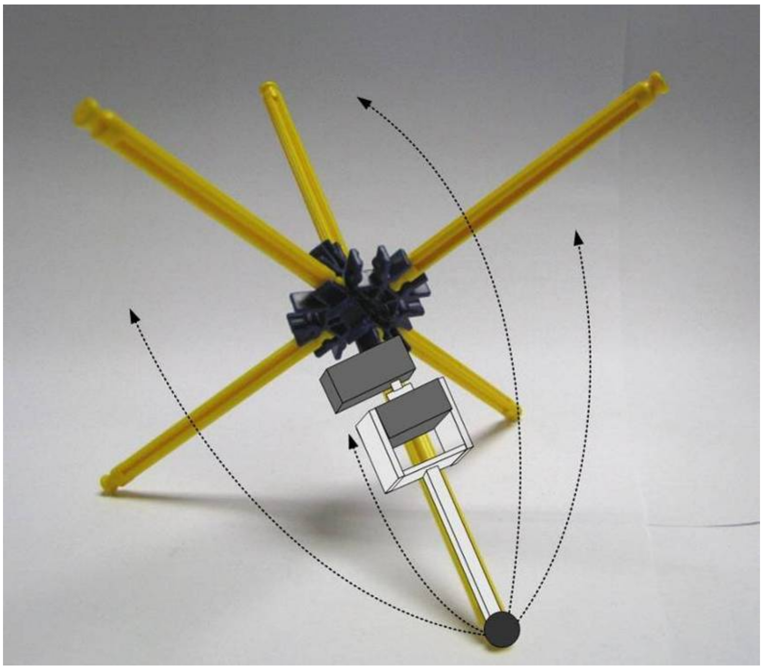


Figure 3-11: Leg Joint Mechanism on Basic Model

As the robot tumbled into new leg configurations, the leg’s relation to the ground altered due to the mounting of the leg’s, motors and brackets. The leg configuration, with respect to the ground, and therefore the triangular region formed by the feet that were placed on the ground, required the base motor to rotate the joint motor into a set position, such that the joint motor was able to then actuate the leg in-line with the direction of travel. The principle of the base rotation as a 'roll' and the joint articulation as a 'pitch' is shown in Figure 3-12.

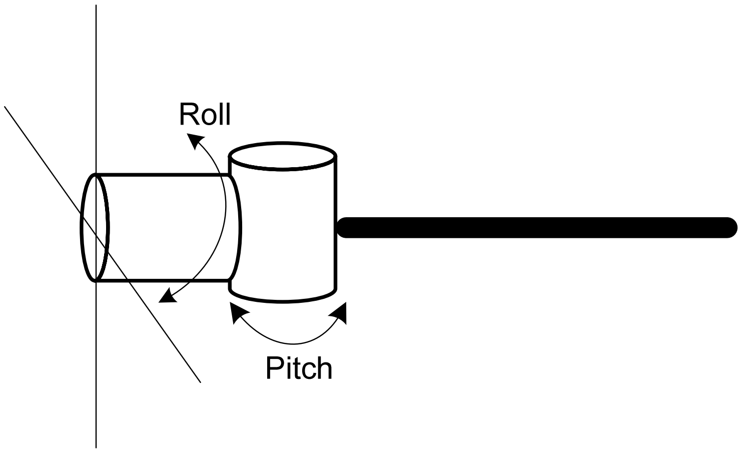


Figure 3-12: Roll and Pitch of Joint

3.4 Summary

Having identified the most effective mechanism to create movement, the following chapters in this thesis will describe the robotic design. The next chapter begins by covering the development process of the robotic hardware. Next is a discussion of the creation of a navigation system, used as a method for obtaining results from the robotic design, and its unique form of motion. Analysis of the navigation system is discussed, and the results obtained from tests allow fulfilment of the research question.

As this thesis addresses the creation of a concept in robotic motion that is unique, the development of application specific hardware is necessary to provide functionality for the robot's unique design and motion. The sound construction of robotic hardware is therefore a critical aspect of the research, to enable the unique form of locomotion to be developed and implemented successfully.

4 ROBOTIC HARDWARE DESIGN

In this chapter, the design of the robotic hardware is described. The robotic hardware consisted of both mechanical and electrical systems. The robot was constructed to form a unique design that allowed a novel and unique form of locomotion to be created. This chapter focuses on the design process and concepts of the mechanical and electrical systems, including a description of the integration of these systems for creation of a fully functioning robot.

4.1 Specification

The purpose of constructing the robotic hardware was to create a device to investigate a unique form of robotic motion. The robotic design used a combination of commercially available components, and numerous custom designed components, designed specifically for this application. As the robotic design is unique, the creation of systems that were able to seamlessly integrate into the design was essential. Custom creation of components and systems allowed the abilities of the robot to be maximised, by exploiting the natural characteristics of the robot's design, to allow construction of a densely complex structure.

The development of the robotic design stemmed from a basic model and hence the fabrication of a chassis and creation of the robot's form could be quickly realised. The robotic hardware design had several objectives:

- Concept
    - Exhibit a unique gait, exhibit a unique design, achieve navigation
- Mechanical
    - Robust structure, light weight, fully integrated, custom design, serviceable, compact
- Electrical
    - Expandable abilities, internal power and processing, wireless communication, high level of integration

4.2 Mechanical Design

4.2.1 Leg Joint Design

The leg and joint mechanism for Spike consisted of two standard brackets, a hub, a limb extension and a foot. The two brackets were configured with two servo motors in a way that created a joint to articulate the limb extension and foot. The joint's freedom of movement was aided by the use of ball bearings where necessary, which allowed the maximum torque of the motor to act against the ground, and hence minimised dissipation in the actual bracket mechanism. The brackets were commercial 'off-the-shelf' type, brushed aluminium pieces that had necessary mounting points pre-drilled for integration into systems using servo motors. The assembly, configured appropriately for Spike, formed the leg and joint mechanism for the robot.

From the chassis, the first multi-purpose bracket attached to the base motor that was mounted to the chassis and was designed to mount the second joint motor. This bracket is shown in Figure 4-1. The second joint bracket was a C shaped design that was used to attach the limb extension to the joint motor. The C shaped bracket configured with the base and joint servo motors, is shown in Figure 4-2. As the drive shafts of each of the servo motors are offset, the rotation points of the overall system are carefully considered, with the bracket and motor configuration used to preserve the central axis about the cube centre.

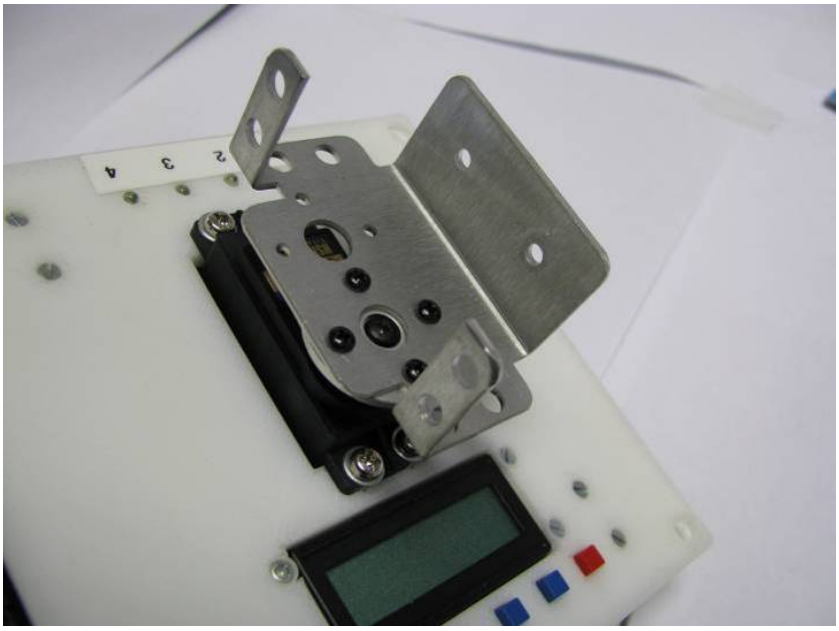


Figure 4-1: First Bracket on Base Motor

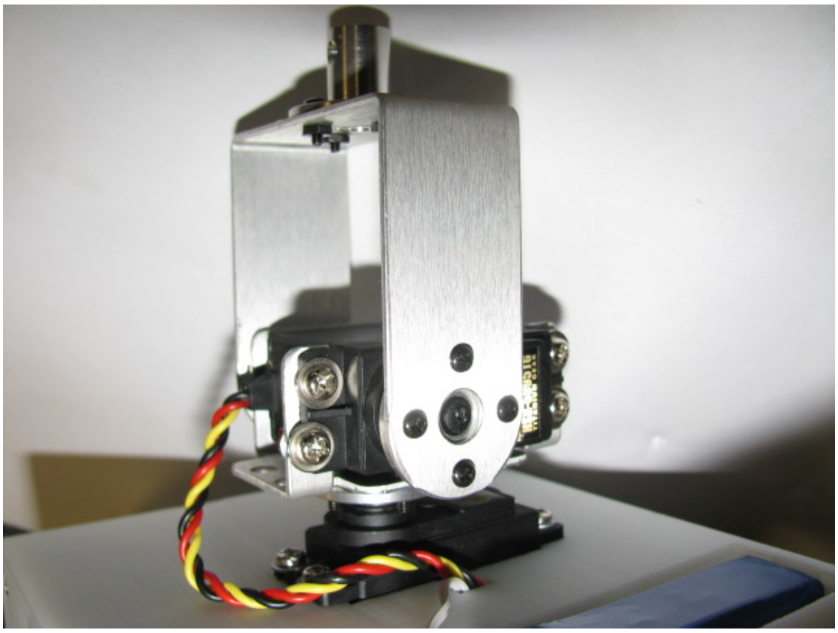


Figure 4-2: Second Bracket with Motors

The limb extension consisted of a hollow aluminium tube which served as the robot's leg, and also provided cable feed ability between sensors attached to the feet, as well as the signalling from the electrical systems of the robot. The leg is shown in Figure 4-3. Each cable attached directly to the LDR (light dependant resistor) light sensors that were mounted within the foot of each of the legs seen in Figure 4-4. The limb extension attached to the C shaped bracket, by use of a hub as an intermediary attachment device.

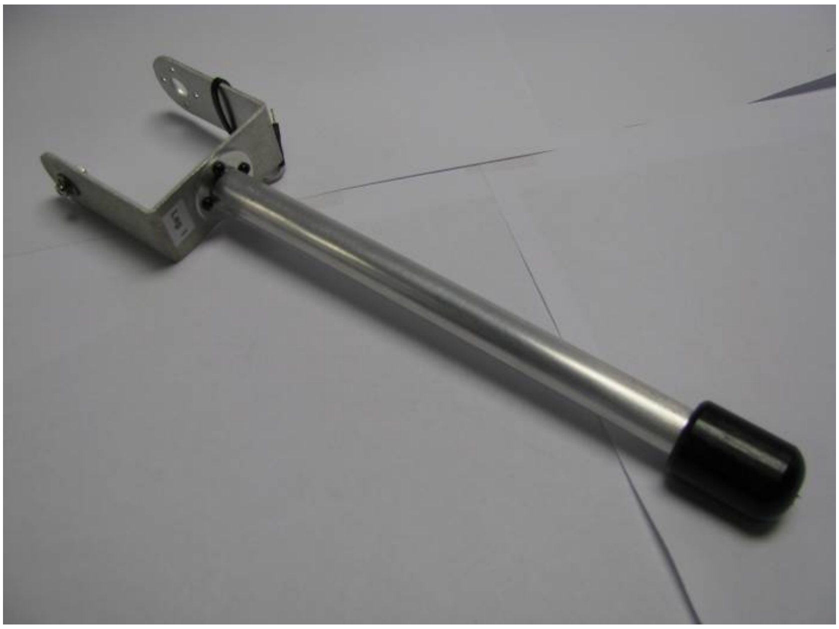


Figure 4-3: Leg

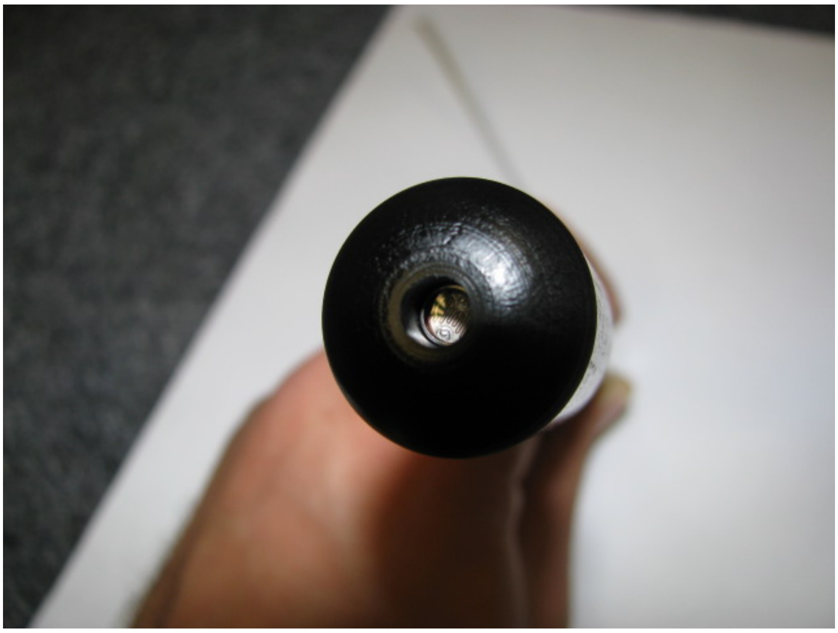


Figure 4-4: LDR Light Sensor Inside Foot

The joint mechanism was powered by two high torque titanium geared servo motors. Each motor had stall torque characteristics of 2.94 N-m, and a standing torque of 3.82 N-m. The configuration of the motors in the joint allowed two degrees of freedom for each leg, consisting of leg rotation and leg actuation. The purpose of the rotation was to correctly position each leg with respect to the ground surface. After correct positioning, the joint then actuated the leg and foot to tumble the robot. Further description of the joint mechanism's range of motion is discussed in section 3.3.

The foot was custom designed as a low friction, interchangeable device that attached to the main leg extension. The foot was designed to slide along the ground surface with minimal friction and reduced wear. It was turned from acetyl plastic on a lathe, and was designed with centre hole and internal lip for mounting the LDR light sensors within each leg. The positioning of the internal lip within the foot allowed the LDR sensor to recede by 5mm from the end of each foot, which assisted the prevention of unwanted stray light, while also preventing damage occurring to the sensor itself.

4.2.2 Importance of Leg Length at Tipping Point

A drawing of the mounting of the leg and joint mechanism to the cube is shown in Figure 4-5. To determine the minimum length required for the driving leg, the point at which the robot was about to tumble was examined as a critical point. The leg length was consequently determined by consideration of the robot's geometry at its tipping point shown in Figure 4-6 and Figure 4-7.

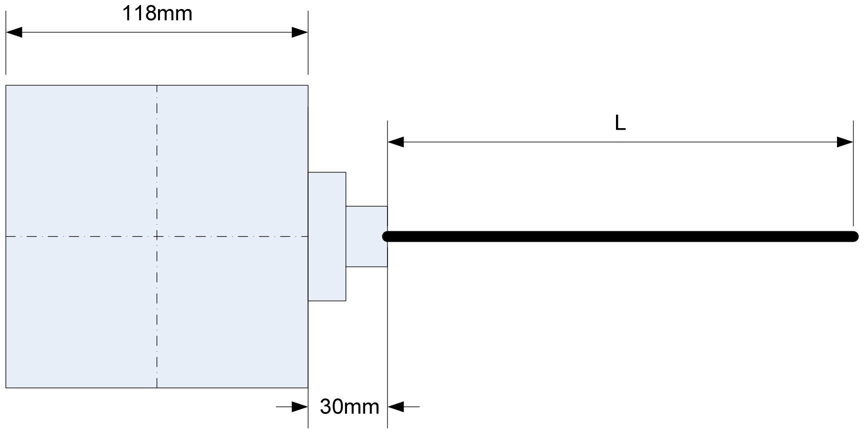


Figure 4-5: Chassis, Joint and Leg Drawing

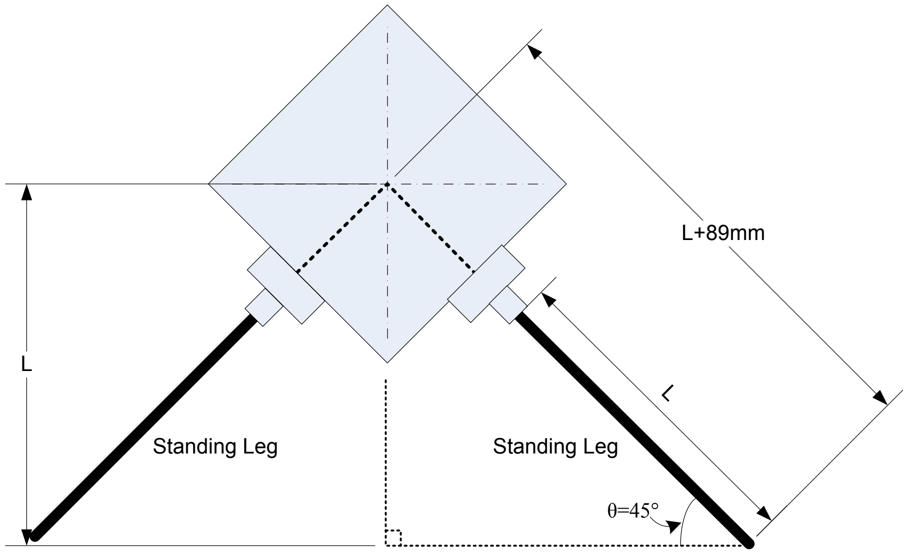


Figure 4-6: Rear View Drawing

The proof for the minimum required leg length to cause the robot to tumble was determined, where 'L+89' is the total distance from the centre of the cube to the end of the foot, and θ is 45° to the ground.

$$(L+89)\times\sin 45^\circ = L$$

$$L = \frac{89\times\sin 45}{1-\sin 45} = 215mm$$

$$(215+89)\times\sin 45^{\circ}=215mm$$

Therefore the leg length L must have a minimum length of 215mm to allow the robot to tumble. As this length was a minimum length, an additional tolerance was included into the leg design of 20mm to ensure robust motion. The final leg length L used in the design was 235mm.

The side view of the robot using the actual robot dimensions and including standing and driving leg, is shown in Figure 4-7. The figure shows the robot at its critical tipping position, and leg length used on the robot to create the tumbling motion.

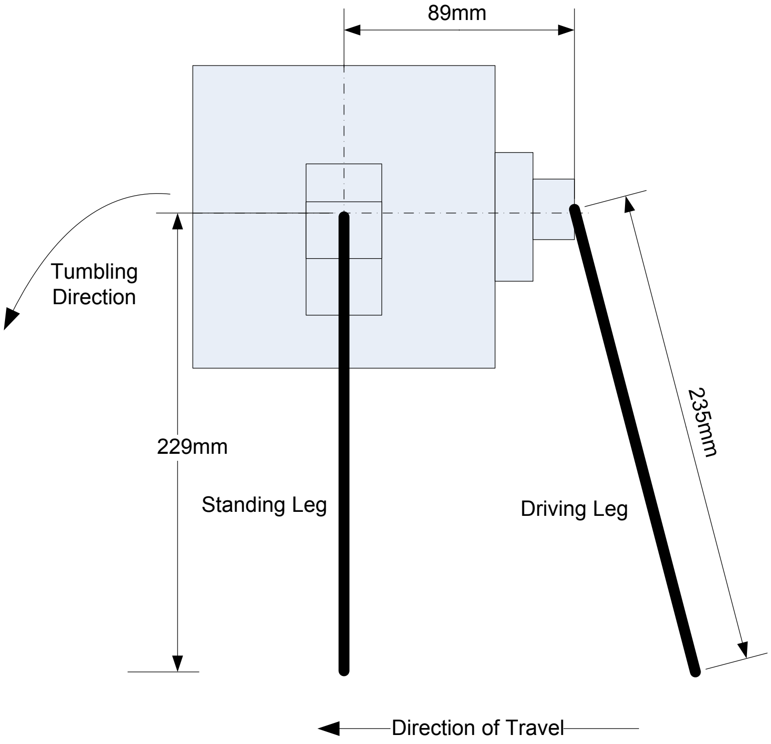


Figure 4-7: Side View of Robot at Tipping Point

From these calculations, it was possible to determine that the vertical height between ground and the centre of the cube must be less than the actual leg's length for the tipping to occur. Using the actual leg dimensions of 235mm specified, the vertical height between the cube centre and the ground was 229mm, thus the leg length was sufficiently long enough to tip the robot. To assist the driving leg's ability to tip the robot, the standing legs had the ability to be adjusted by the servo motors to decrease the vertical height between the ground and centre of cube. This was achievable by adjusting the two joint motors of the standing legs to cause the robot to 'sit' and therefore cause the driving leg's length to have a greater tipping effect.

4.2.3 Chassis Design and Fabrication

The chassis design consisted of a cube-shaped box that included internal compartments and internal features, shown in Figure 4-8. The cube was comprised of two main parts; a lid, and the remaining body. The lid was a single face of the cube that was removable, to provide access to the internal body of the cube. Four compartments included within the body provided a suitable holding bay for batteries, with appropriate spacing for cable feed and clearance for battery removal. The external faces of the cube body featured detachable access lids that provided accessibility to batteries stored within the cube for replacement. The cube was designed to provide a high level of strength by including internal ribbing on all internal joining sides. Ventilation grills providing airflow were included for heat dissipation. The design included all necessary mounting points for servo motors, and circuit boards, and all necessary holes to provide full integration of the electrical components and the robotic chassis.

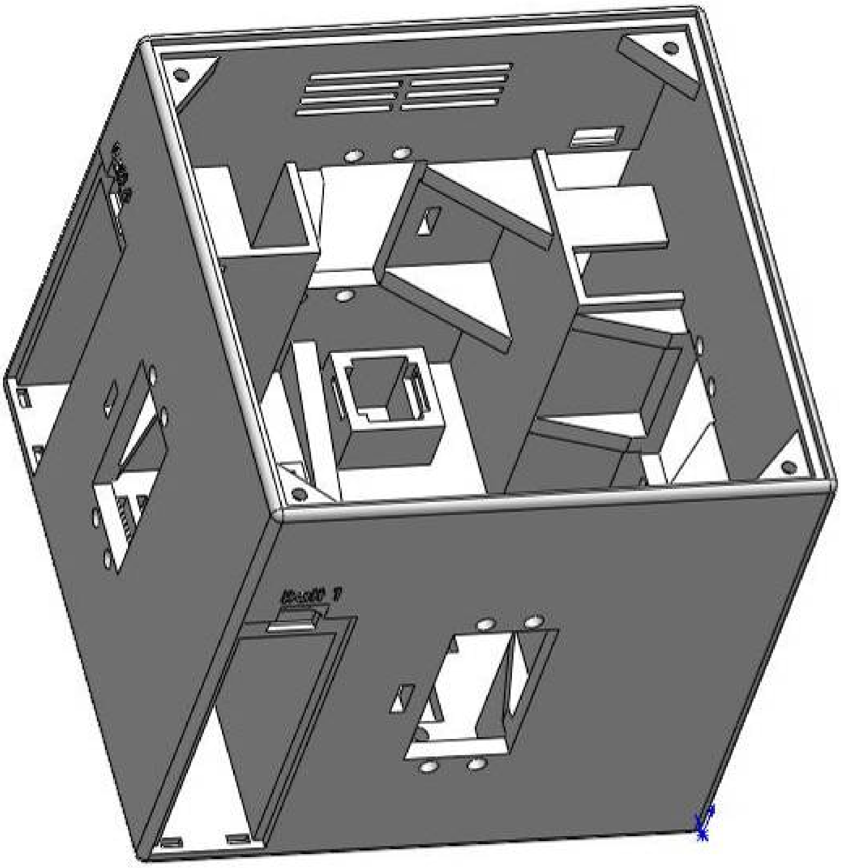


Figure 4-8: The Cube Body Design

Initially, the sketches of the chassis design were hand drawn, and were later computerised into SolidWorks CAD software. The use of CAD software provided the ability to model the compartments and internal free space of the cube, to allow the other hardware components to fully utilise the free space of the cube. This allowed the abilities of the robot to be maximised with efficient use of the cube's free space. The CAD software also allowed small details to be easily manipulated such as: including countersink cavities for screw heads for a professional finish, and the rounding of corners and etched lettering to be incorporated as part of the cube design.

The cube design was fabricated using a 3D plastics rapid prototyping printer. This manufacturing process allowed the creation of physical objects to be built up in layers from plastic material, to form solid objects. By manufacturing the cube in plastics using a rapid prototyping printer, many advantages were seen. This allowed the cheaply and precisely manufactured cube shape to include the previously mentioned complex internal structures. The chassis had sufficient strength, suitable for the impact experienced with each leg movement, and the robot's tumbling motion on the ground.

A conventional method to fabricate the cube may have been to craft the chassis design from metal or some other material with the aid of a highly skilled tool maker. The benefit of creating the design from another material may have seen advantages in strength or weight characteristics when compared with the rapid plastics prototyping design approach. This would undoubtedly produce a quality end result, however the development time and cost of production would likely stretch beyond the scope of this development. The many internal features of the body would also pose challenges for conventional construction methods, because the production run would consist of only one unit. In Spike's development, the benefits of using a rapid prototyping printer were very evident and therefore rapid prototyping printing was the only realistic option.

The battery lid shown in Figure 4-9 included tabs for fixing the base of the lid to the main cube body, and a flexible clip system that allowed the lid to be attached or removed from the cube. The clip mechanism was designed in such a way as to permit flexing in its joint, allowing the catch to be released from the main body, and thus the ability to remove or replace the lid. The lid was mounted on the main body in a recessed region, to allow a flush finish on the chassis's exterior.

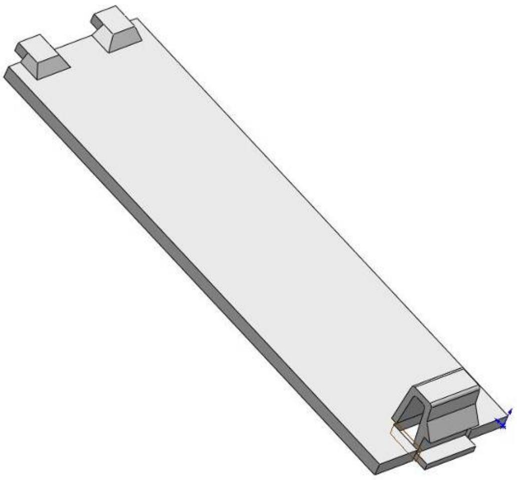


Figure 4-9: Battery Lid Design

The main lid from the cube body is shown in Figure 4-10. This lid was mounted to the main body via four screws located at each of the corners of its body, and was seated on a recessed lip within the main cube body. The lid used cut out holes for servo motor and LCD (liquid crystal display) screen placement, and several mounting holes for attaching circuit boards located within the box. The drill hole placement for mounting buttons and LED (light emitting diode) lights was designed for correct alignment with the lid's centre and the relative positions with respect to the LCD screen. Text on the surface of the lid was included to indicate necessary functions of the features.

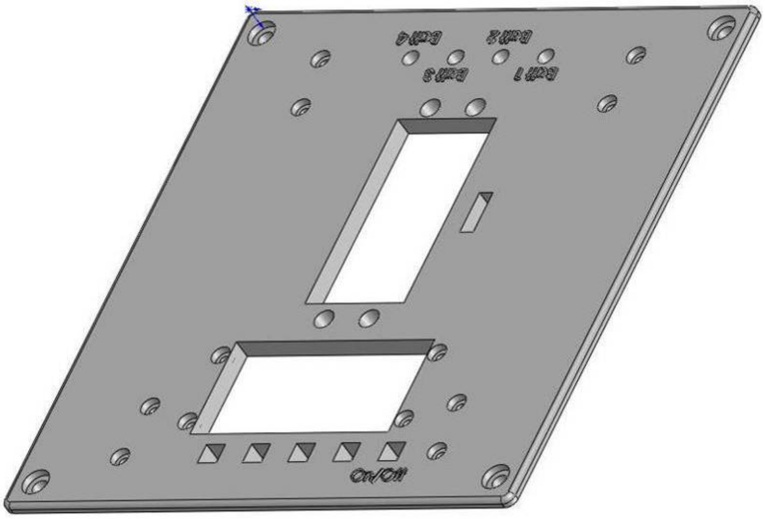


Figure 4-10: Main Lid Design

By including the features of the main body and lid, the ability to create a professional and finished feel for the design was achievable. It allowed a balance between function and form which allowed for the full integration of all components required by the robot.

In Figure 4-11 the lid is shown mounted to the main body with relevant components attached in its final assembly.

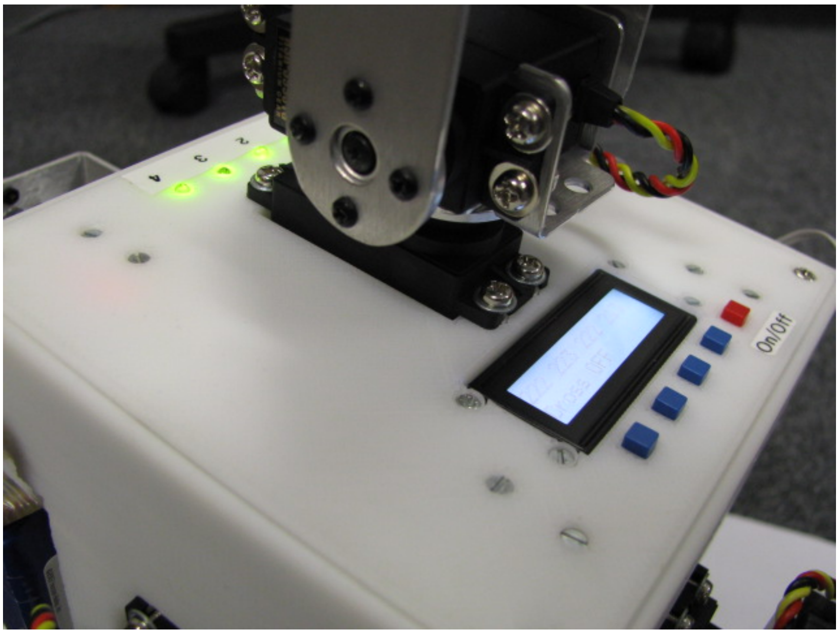


Figure 4-11: Lid Mounted to Main Body Including Components

The dimensions of the cube's construction were considered as a trade off between expansion of the electrical abilities of the robot, and creating a small and portable design. By creating a larger internal space within the cube, the ability to include a greater number of processing and electrical features increases, but the overall size, weight and power that are integral for the robot's operation also increase. The final dimensions of the cube are summarized in Table 4-1.

Internal Dimension	110mm
External Dimension	118mm
Wall Thickness	4mm

Table 4-1: Cube Dimensions

4.2.4 Torque Calculations

The robot was inherently stable at rest, as its centre of mass was balanced in equilibrium by its tripedal stance, formed using its three axes of symmetry. The forces acting on the robot legs are shown in Figure 4-12. The robot weighed approximately 2.2 kg and its height varied throughout the tumbling motion.

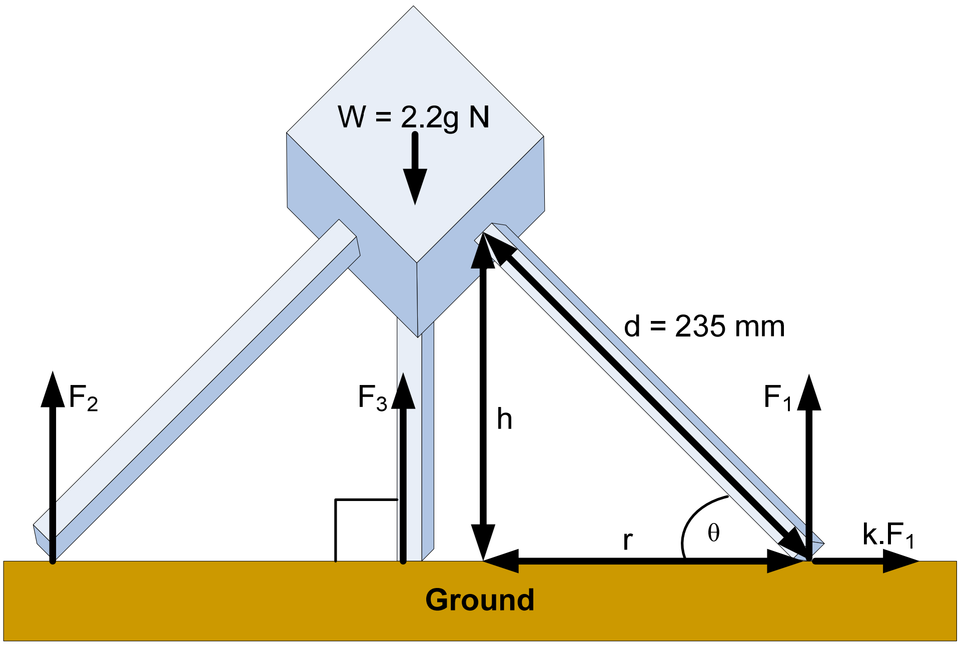


Figure 4-12: Forces on Legs

To provide motion for the robot, high torque capable servo motors were used to drive the legs. To ensure the motors had sufficient torque ability to force the legs to shift on a carpeted surface, the maximum torque acting on the legs was determined as the opposing turning moment acting on the servo motor.

Mass	2.2Kg
Leg Length 'd'	0.235m
Number of Legs	3
Friction co-efficient of ground 'k'	0.7
Gravity 'g'	9.81

Table 4-2: Table of Robot Specifications for Torque Calculation

To determine the maximum opposing torque, the opposing force acting on the leg was multiplied by the perpendicular distance.

The vertical height ‘h’ is:

$$h = d\sin(\theta)$$

And horizontal distance ‘r’ is:

$$r = d\cos(\theta)$$

At each servo, the moment due to the weight reaction force is:

$$\frac{W}{3}r$$

And the moment due to horizontal friction force:

$$\frac{W}{3}kh$$

The total moment of each leg:

$$\tau = \frac{W}{3}\left(r + kh\right)$$

$$\therefore \tau = \frac{W}{3}d\left(\cos\theta + k\sin\theta\right)$$

So to calculate θ when τ is a maximum:

$$\frac{d\tau}{d\theta} = 0$$

$$-\sin\theta + k\cos\theta = 0$$

$$k\cos\theta = \sin\theta$$

$$\tan\theta = \frac{\sin\theta}{\cos\theta} = k$$

So for different values of k we have in Table 4-3:

k	θ
0.1	5.7°
0.2	11.3°
0.3	16.7°
0.4	21.8°
0.5	26.6°
0.6	31.0°
0.7	35.0°
0.8	38.7°
0.9	42.0°
1	45.0°

Table 4-3: Maximum Opposing Torque Angles for Different Surface Friction Coefficient's

∴ As the friction of the carpet in the test situation is k≈0.7, the maximum opposing torque on the servo motor is:

$$\therefore \tau = \frac{2.2 \times 9.81}{3} \times 0.235 \times (\cos 35 + 0.7 \times \sin 35)$$

$$\tau = 2.06Nm$$

The two servo motors used in this application were capable of delivering 30kg.cm (2.94 N-m) of torque at stall, and 39kg.cm (3.82 N-m) as a standing torque, and were therefore capable of overcoming the effect of the opposing turning moment experienced by each leg at its worst angle position. A total of six HSR-5995TG servos were used as base motors, and six HSR-5990TG servos were used as the joint motors. The HSR-5990TG servos incorporated a chassis that included heat sink that assisted with heat dissipation. As the joint motor provided the leg actuation, and was therefore required to perform work against a greater resistance in comparison to the base motor, the heat dissipation characteristics of the actuation motor were better suited in comparison to the HSR-5995TG servos that did not feature a heat sink. The use of non heat-sink motors represented a cost saving in the robotic design.

4.3 Electrical Systems

4.3.1 Overview

This section provides a description of the electrical systems developed for this robotic application. Specialised custom designed electronics, purpose built for Spike were required to allow the successful creation of a wireless robotic platform. The custom electrical design allowed the abilities of the robot to be maximised, by effective use of the free space contained within the cube chassis. Effective use of the internal space required the design to be built vertically through the cube's centre where possible. The final assembled boards stacked for insertion into the cube are shown in Figure 4-13.

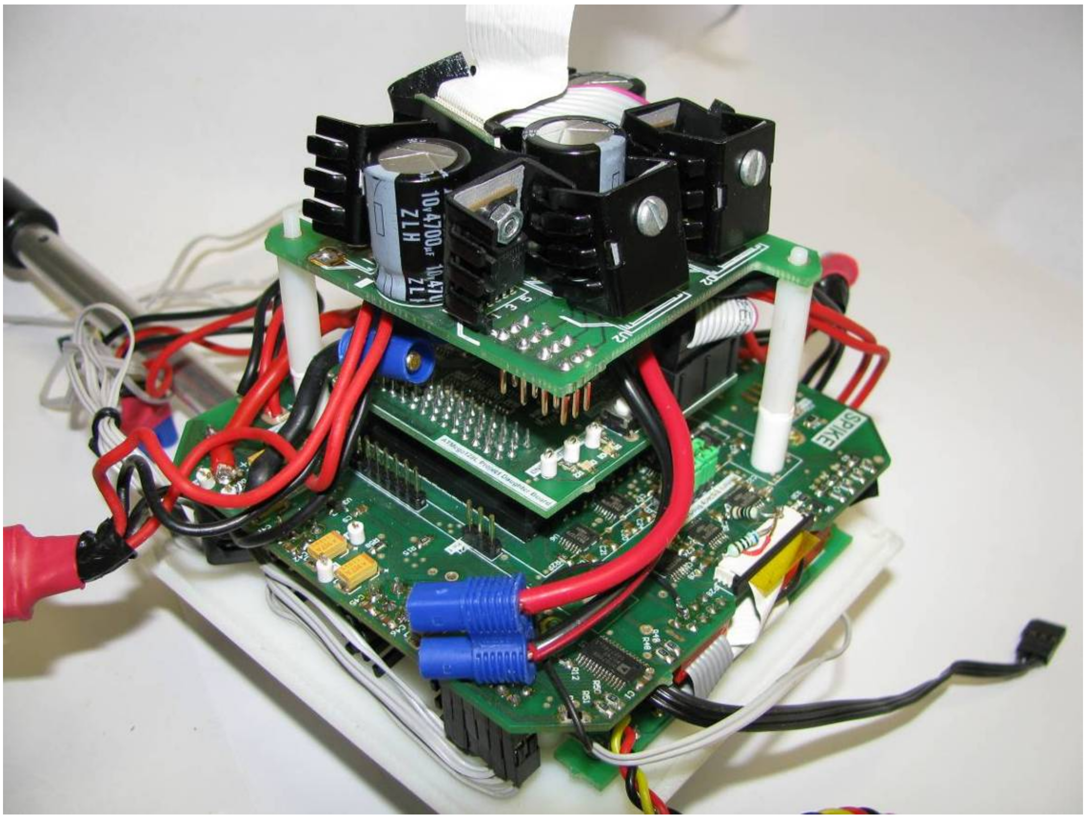


Figure 4-13: Stacked Circuit Boards

The electrical system was designed to interface to external components such as; the servo motors and analogue sensors, and to process this information appropriately to facilitate the robot's control. Although many separate commercially available 'off-the-shelf' components might have been suitable to perform a small selection of tasks required by the robot, the densely complex cube design demanded custom fabrication to integrate all systems within the unique internal chassis space.

The electrical overview of the robot is presented in Figure 4-14.

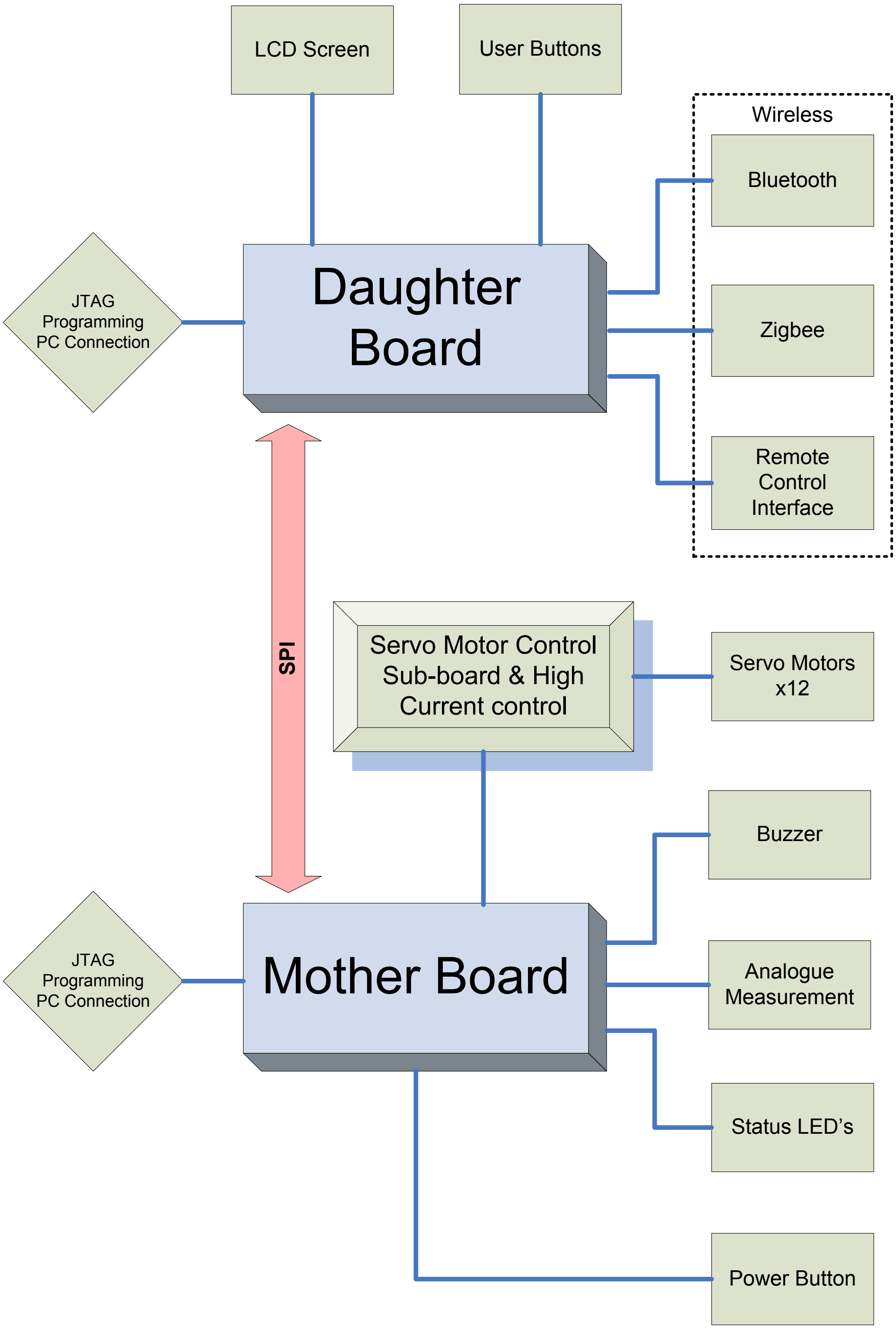


Figure 4-14: Spike Electrical Overview

The design of electrical systems in the robot involved both schematic and board routing design to create several dual-layer printed circuit boards for the robot. Significant time was spent at this electrical design stage to successfully create a sophisticated electronic design that allowed the robot to function as a fully operational and powerful robot.

An important aspect of the electrical design was the implementation of power switching using MOSFETS. The power for components and the entire robot system could be toggled via the electronics and software programming, therefore careful attention was required to ensure that correct design of low side and high side switching was achieved. The power switching technique allowed the goal for overall soft power switching control to be achieved. The power switching designed in this way also allowed every component's power supply in the robot's circuits to be individually controlled for added customisation and control.

The main processing and control circuit design involved the creation of two distinct circuit boards noted as high and low level control. High level control consisted primarily of navigation abilities and decision making and was processed using the daughterboard. The low level control consisted primarily of servo pulsing and data collection and was performed using the motherboard. Additional circuit boards were implemented in the design that each performed a unique task. The servo and high current control board was responsible for the management and distribution of power to high drain devices such as servo motors. The buttons, LCD and LED's each used separate circuit boards for custom attachment to the lid panel of the chassis and were electrically connected via ribbon cable to their controlling boards, and each wireless module was also attached separately to the motherboard via a connector. The following section will highlight the important features of the robot's electrical design. The schematic and PCB (printed circuit board) diagrams are included on the attached CD-ROM.

4.3.2 Robotic Communications

4.3.2.1 Remote

The wireless remote control device used in the project was a Sony Playstation® 2 styled remote controller. The device was designed for use with the Sony Playstation®, but with appropriate modification of its receiving interface, direct implementation into the robot design was achievable. Using the remote control allowed the user to interact with the robot using up to 14 buttons and 2 analogue joysticks. Use of an analogue joystick was especially useful when selecting the desired heading for the robot to travel toward, as the joystick provided a meaningful position feedback to the user that related to the desired heading for the robot to follow. The decision to use this remote control was assisted by the many features it possessed and its analogue control ability, allowing harmonious integration into this robotic application.

As the remote control device was not specifically designed for this application, significant modification and development of its interface was required to unite its abilities with the robot. To facilitate the communication, four I/O lines were established:

- CLK
    - Clock line, used to synchronise the receiver. A data bit was shifted on every falling edge of the clock generated from the daughterboard processor.
- CMD
    - Command line, in which data was sent from the daughter processor to the controller. This was normally high.
- DATA
    - Data, received from the remote control. This was normally high.
- ATT
    - Attention, used by the daughterboard to activate communication with the remote controller.

The remote control is shown in Figure 4-15.

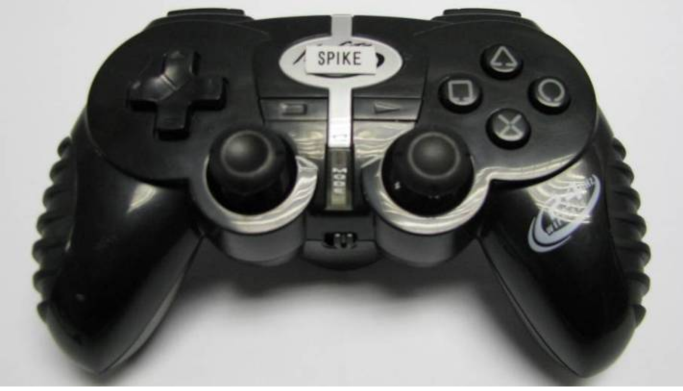


Figure 4-15: Remote Controller

4.3.2.2 Bluetooth

Bluetooth is a wireless protocol used to send and receive data over a short distance. It has become a standardised protocol for creating wireless communication links between personal devices such as cell phones and computers. Bluetooth was used in the Spike robot to allow computation data to be transmitted from the robot as it moved in its environment. Due to the fact that the robot exhibited a tumbling motion, the use of a tethered cable was not practical as the robot would quickly become entangled. A standardised 'off-the-shelf' class 1 bluetooth modem module was purchased and implemented into the robotic design.

As bluetooth modules are found in most modern computers and devices, implementing this wireless protocol opened a multitude of possibilities for connection of PC's and other devices to the robot. Another benefit of using a bluetooth wireless module was development with a commonly known, standardised platform that allowed rapid integration of its many features into the robot system. The bluetooth module is presented in Figure 4-16.

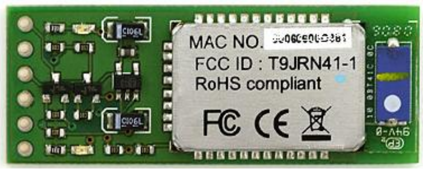


Figure 4-16: Bluetooth Module

4.3.2.3 Zigbee

The future expansion of Spike's control system has the potential for development with evolutionary based control systems. Potential future development of an evolutionary control system for the robot requires a method for the robot to determine its actual displacement, and a method to determine slippage by the feet. This ability may be included in the robot with the use of zigbee wireless technology with an inbuilt location engine.

The principle of the location engine concept using the zigbee module was to use reference nodes placed in the robot's environment, where the distance between the reference nodes is known. The zigbee module located within the robot is then able to determine its position with reference to these nodes by measurement of wireless signal strength, and transmit its coordinate as an (X,Y) value. This may be used to determine the actual robot position and evaluate the effectiveness of the evolved control system.

The location engine may also be used to allow the robot to take corrective actions where the slippage of the robot over a number of steps causes an error from its mathematically calculated position. The zigbee development board using the Chipcon (by Texas Instruments) CC2431 IC with included location engine detector is shown in Figure 4-17.

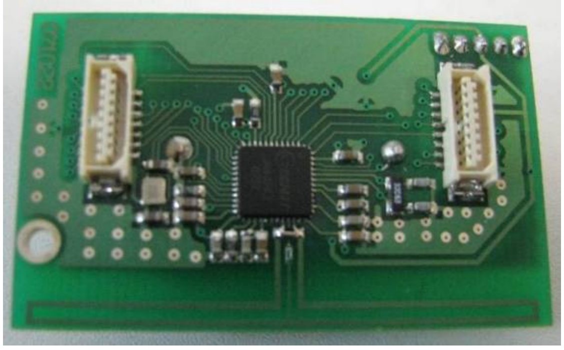


Figure 4-17: Zigbee Module

4.3.3 Sensors

The Spike robot is a three dimensional object that moves in space, and has no natural or inherent orientation in its environment. To allow the legs to be manipulated in a controlled manor to achieve motion, the robot required sensory devices to allow its control circuitry to determine its orientation. This was achieved at the robot's design stage in two ways, using light level measurement and position displacement. The position displacement technique that might be used via the zigbee module's location engine is discussed in Section 4.3.2.3 and Section 6.2.3.3. This aspect of the robot requires further development.

The light level measurement was achieved with the use of light dependant resistors mounted within the foot of each leg. LDR devices were selected over other measurement systems for their robustness, simplicity, cost effectiveness and ease of integration into Spike. The use of these sensors allowed the robot to draw a comparison between its leg positions, and with reference to its natural symmetry to determine how the robot was orientated in its environment. To achieve this, a 'voltage divider' circuit was designed where the LDR was used to directly influence the output voltage of the divider. This output voltage was then measured and used by the internal processing of the robot. A high voltage indicated a higher level of light, and a low voltage indicated a lower level of light. This was later used to determine the robot's orientation.

The positioning of the LDR in the voltage divider circuit allowed various other sensors (such as foot pressure sensors) to be interchanged if desired. The sensor connection was of a standard header type, and allowed each type of sensor to be easily swapped to other types if required in the future. The LDR sensor is shown in Figure 4-18.

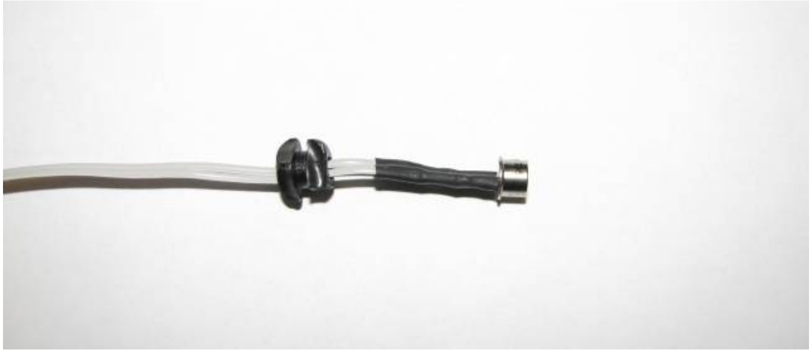


Figure 4-18: LDR Light Sensor

4.3.4 Power Systems & Switching

The power supply for the robot consisted of a dual input supply selection. The dual supply for the robot was from either lithium polymer batteries located within the chassis, or a dedicated external DC input voltage source. The purpose of using an external DC input source was to allow the robot to be programmed and left powered on for extended periods of time on a testing bench. The included batteries allowed the robot to operate without any attached cables when being used for evaluation. This input voltage selection was achieved using a dual input diode, where the input source supplying the highest voltage was selected as the primary source. For Spike's design, this allowed the DC source to be used when available, otherwise batteries were used.

Power delivery with batteries was split among four separate lithium polymer batteries. Each battery was 7.4VDC 1250mAh with a maximum discharge rate of up to 20 times is current capacity (20C). This provided the ability for each battery to deliver up to 25A if required under extreme loads. A single battery was dedicated for powering the discrete electronic circuits within the cube, and the remaining three batteries were split evenly between the twelve servo motors. This configuration allowed smaller batteries to be used in the design that provided space and weight saving, whilst allowing the robot to remain mechanically balanced with even weight distribution around the chassis. A diagram of the electrical power connections is shown in Figure 4-19.

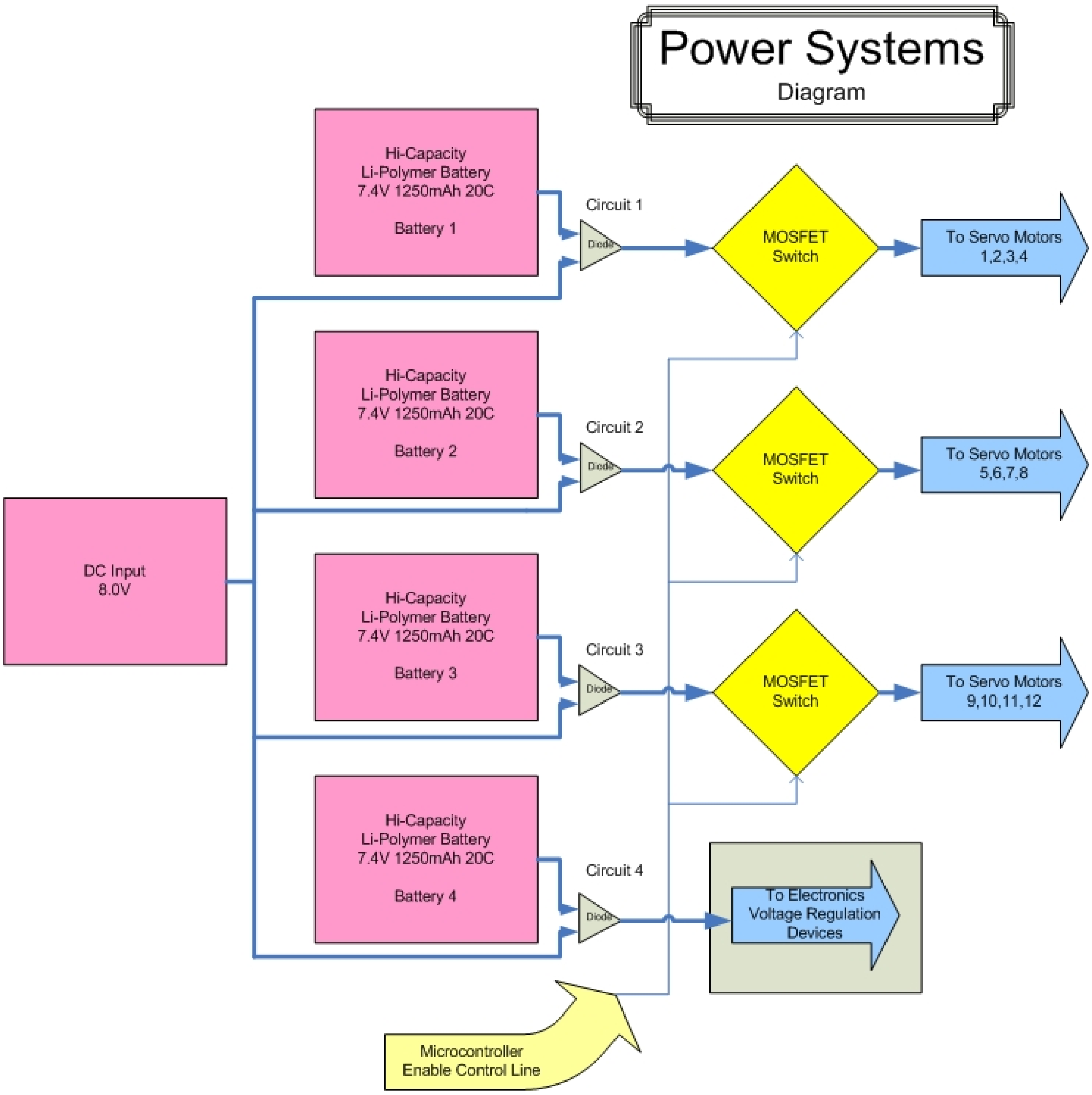


Figure 4-19: Robot Power Sources Diagram

The connection and switching operations were performed on a dedicated power switching and distribution circuit board located within the cube. The purpose of building this board was to separate all high current devices and switching abilities from the remaining low power discrete components located on remaining circuit boards. The various terminal connections for power supplies and motors were included on the high power board where real estate was available, created by separation of the power systems from the motherboard.

Switching of the three motor supply batteries was performed using high-current intelligent MOSFET devices that acted as high side switches. The switching of these

MOSFET's was controlled by the motherboard microcontroller. This allowed the robot to selectively enable these high current devices at its discretion, to act as an On/Off switch for the motor supply. The board is shown in Figure 4-20.

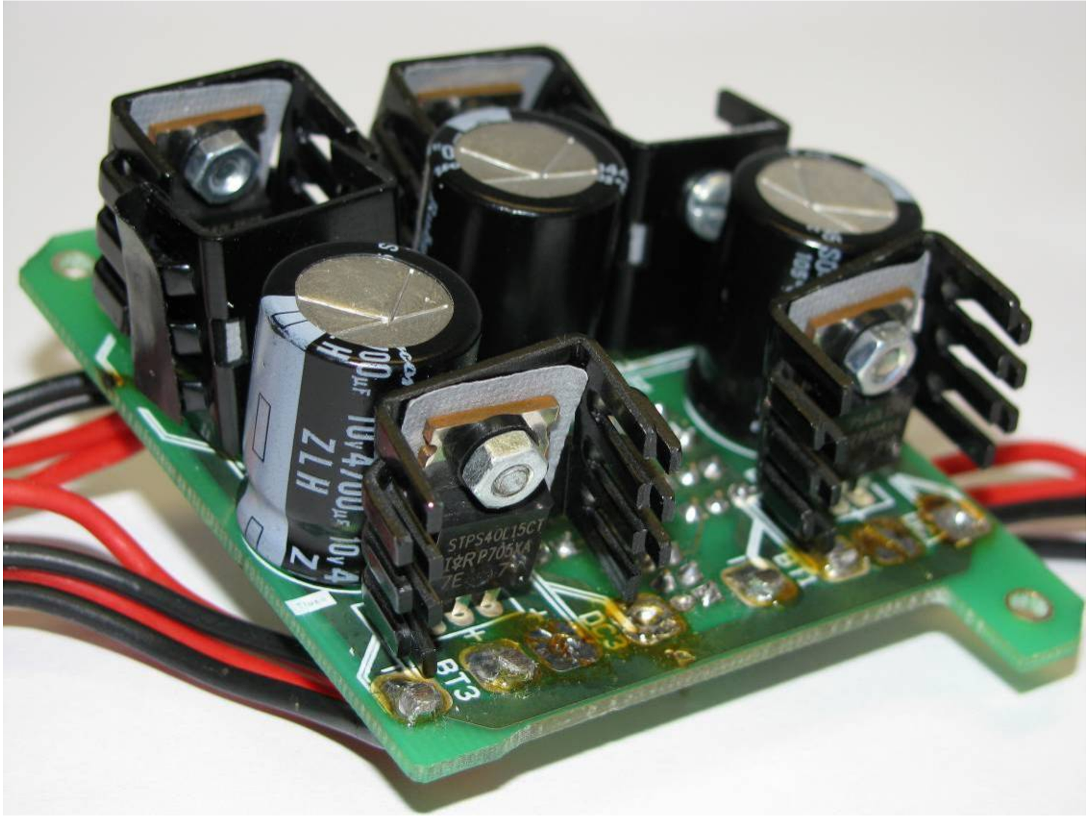


Figure 4-20: High Current Power Board

To allow the high side switches to selectively power the servo motors, the power supply for the microcontroller required a separate dedicated power switching method. This method is shown in Figure 4-21 where the microcontroller used a dedicated regulation source to maintain power for itself. The remaining discrete components then fell under control of the microcontroller's switching ability. If the robot was required to be powered off, the motherboard microcontroller entered a low power sleep mode, discussed in section 6.3.1 of this thesis.

Individual components of the robot were controllable via a series of low side power MOSFET switches. By controlling the individual power to devices such as buzzers, LCD back light and wireless modules, the individual components of the board were under direct control of the running program on the microcontroller. This method allowed devices not in use to be powered off by the running software to conserve battery power.

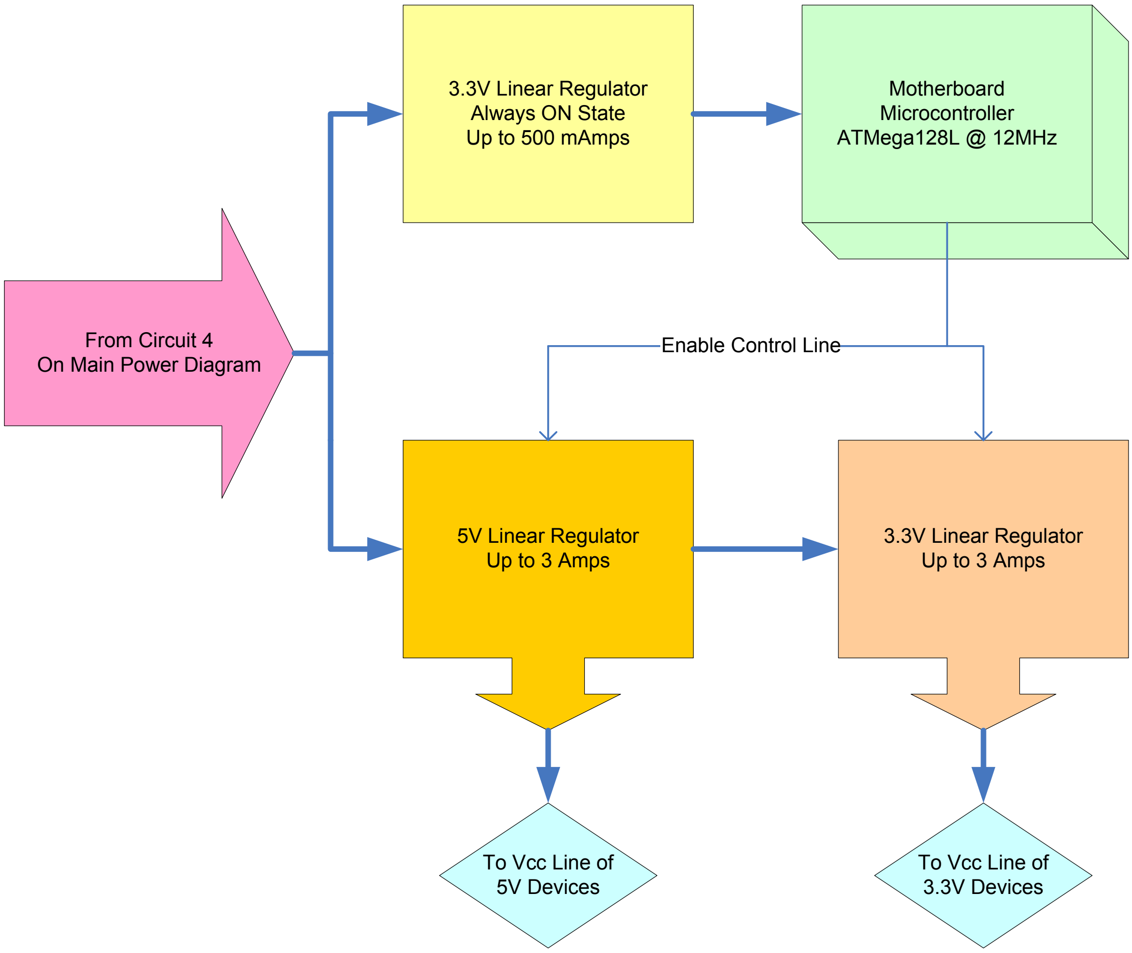


Figure 4-21: Motherboard Only Power Switching Diagram

4.3.5 Motherboard

Spike's motherboard required considerable planning and design before a usable final board was fabricated. The motherboard not only acted as the primary interface for most controlling systems used within Spike, but also as a mechanical fixing point for most electrical circuits within the chassis. The motherboard itself was directly attached to the lid of the enclosure with screws, and remaining devices within the cube were attached to the motherboard. By designing the electrical circuitry in this manner, removal of the chassis lid provided instant serviceable access to the internal devices located within the cube. The populated motherboard is shown in Figure 4-22.

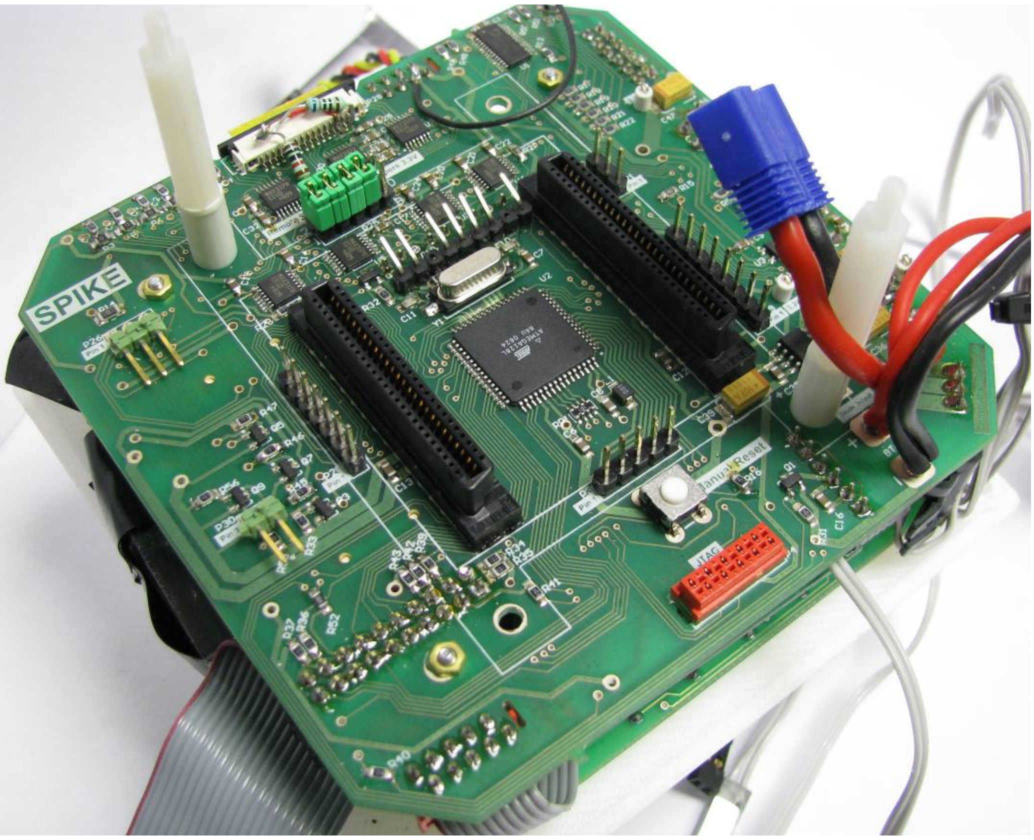


Figure 4-22: Motherboard

Careful attention to the speed requirements for the motherboard systems was critical to the successful operation of the robot, as the time taken to process time critical activities was significant. Examples of tasks included the creation of critical pulse timing given to the 12 servo motors simultaneously, reading and performing calculations on battery voltages and temperature measurement, interfacing to status LED's and buzzers and inter-processor communication. To perform this processing, an Atmel 8-bit RISC AVR Mega128 microcontroller was used. Prior knowledge with programming these microcontrollers assisted with its smooth incorporation into the design.

The motherboard used two separate voltage rails to supply the various onboard devices. In some instances, both supply voltages were made available to components, under the control of the microcontroller. Although most components made use of 3.3V, a selection of devices used 5V, and consequently voltage translation for communication was required. For protection and power off conditions, the translation devices also provided the ability to tri-state all I/O lines for protection of components on each voltage side of the translator. During power-off, both the voltage rails were made unavailable, and the tri-state selection was held activated, ensuring full protection of the onboard components.

4.3.5.1 Servo Motor Control

A commonly used motor in robotics is a servo motor. Servo motors differ from regular DC motors by their ability to provide a precise angular position control that is typically limited to 180° maximum rotation in robotics, but can be modified for continuous rotation if desired. Servo motors used in radio controlled toys are self-contained electromechanical devices that include a feedback circuit, potentiometer and a geared transmission driven by a small DC motor. Servo motors are also used in remote control devices such as positioning the boom of a remote control yacht, and steering in electronic toy cars and aeroplanes. Their use is even seen in modern cars where cruise control systems may utilize their ability to provide tension at a precise point. The internal feedback system used in servo motor designs allows application in systems that require absolute positioning of the motor shaft for control. A conceptual diagram of the servo system is shown in Figure 4-23.

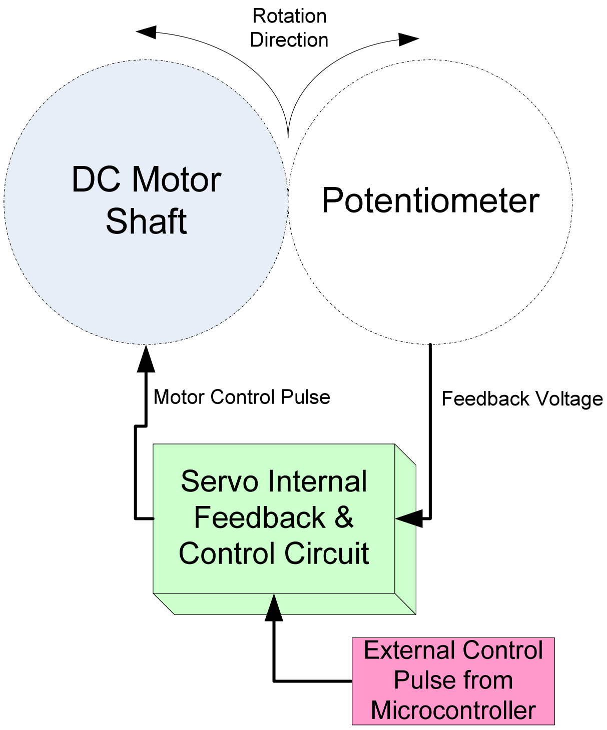

Rotation Direction
DC Motor Shaft
Potentiometer
Motor Control Pulse
Feedback Voltage
Servo Internal Feedback & Control Circuit
External Control Pulse from Microcontroller

Figure 4-23: The Internal Operation of a Servo Motor

The responsibility of the external pulse control circuitry, typically seen in the form of a microcontroller, is to prepare and output control pulse signals to the servo motor that dictates the position of the servo motor's output shaft. In the Spike robot, this was provided in the form of PWM (pulse width modulation) where the 'on time' of the pulse dictated the angular position of the motor shaft. The PWM method is discussed further in section 6.3.2 of this thesis.

Two types of servo motors were used in the Spike robot. These were the HSR-5995TG and HSR-5990TG. The properties of both motors were proven mathematically to provide sufficient torque to drive the robot, and the HSR-5990TG included an aluminium heat sink as its chassis to more effectively dissipate heat. The HSR-5990TG servo motors were a more expensive motor so these motors were only used on the joint that provided the main actuating of the driving leg, and experienced the greatest amount of work, and therefore generation of heat. The twelve servo motors used on the robot are shown pictorially in Figure 4-24.

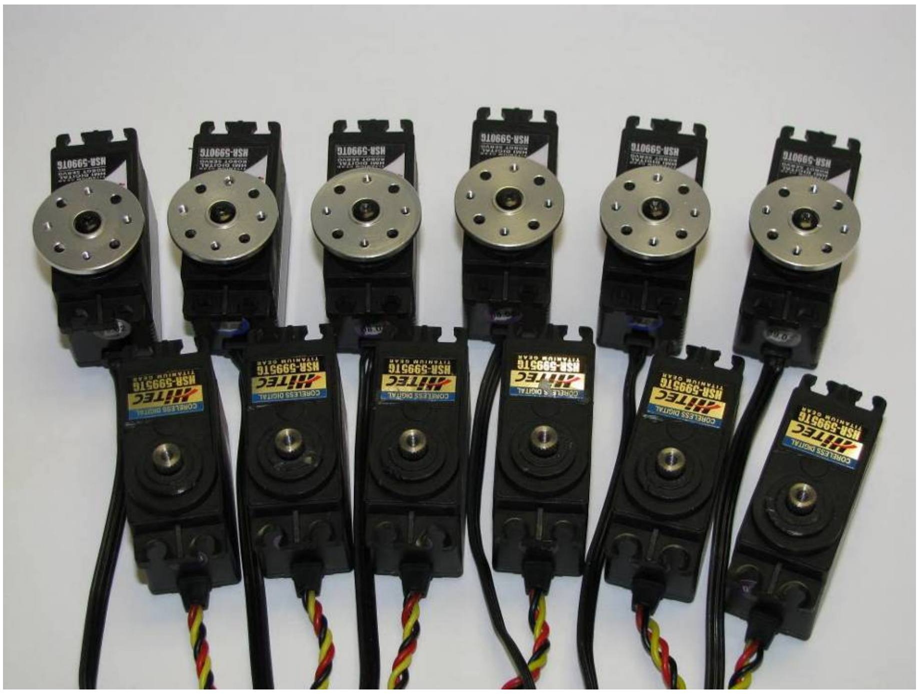


Figure 4-24: Twelve Servo Motors used in Spike

To operate a servo motor, three wires are typically used that provide power, ground and control signals. This allows the power for the servo to be taken from an external power source such as a battery, and the control pulse of its position from a low current microcontroller. The connection of the servo motors was made to the high current control circuit board that provided the necessary power switching and physical connections for the motors. The control pulses sent to this board were connected via a 20 pin ribbon cable to the motherboard, and allowed the high number of connections to be made in a more accessible space within the cube.

4.3.5.2 Temperature and Current Protection

The motherboard electronics design included a method to protect components from thermal overload. This was achieved with use of an onboard temperature sensor, combined with diagnostic fault indication from the intelligent MOSFET switches. Temperature and diagnostic sensors were read frequently to ensure the running conditions for the robot were not exceeding their electrical specification. Eventually, it was found that the favourable running conditions of the robot meant these extra precaution devices did not need to be continually monitored.

4.3.6 Daughterboard

The design of the daughterboard allowed an easily interchangeable processing board to be adapted into any generic system. This was achieved using connectors that allowed an Atmel Mega128 microcontroller to be 'piggy-backed' into an existing system. By creating an interchangeable interface, additional control boards may use different processing technologies and be included into any system. The standardised pin configuration for the daughterboard followed a common pin configuration used throughout AUT University, and thus the daughterboard was able to connect to other devices at the university. The Atmel microcontroller was selected due to a desire to conform to a standard board specification used at the university.

The board included use of current limiting resistors for I/O protection, debug test points and debug LED's, a manual reset button, programming connection and 8MHz quartz crystal. The daughterboard is shown in Figure 4-25.

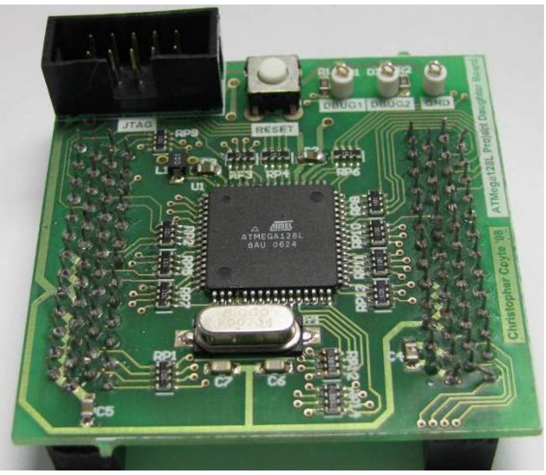


Figure 4-25: Daughterboard

The purpose of the daughterboard in Spike was to assume the high level control for the robot system. The board was physically small measuring 50mmx50mm and hence, the design allowed the vertical stacking of circuit boards within the cube to be achieved.

5 NAVIGATION SYSTEM

5.1 Overview

The functionality of the robot was facilitated by the use of specialised hardware as discussed in previous sections of this thesis. When constructed, the robot required a gait that could be repeated to form multiple steps that would allow an evaluation of the robot's ability to be performed. To allow results and measurements to be taken from the robot, a navigation system was developed in software and programmed into the processors of the robot. The navigation system provided the ability for the robot to receive a destination goal from the user, and translate the goal into an initial sequence of leg movements that the robot would take in order to achieve that goal. For the navigation system to function, the robot required:

- A goal that was provided in the form of a desired heading
- An ability to select legs and motors to create motion, based on the robot's orientation
- Continuous calculation of its current position from an initial configuration

The desired heading was received at the start of the motion sequence from a remote controller that was adjusted by the user. The values received from the remote controller were interpreted and converted into a compass type trajectory for the robot to follow. The desired trajectory was displayed on the robot's LCD screen in real time and was a relative to the robot's starting position. The job of the navigation system was to determine where the trajectory lay around the robot's footprint, and base a selection of leg sequences on that trajectory. This allowed the robot to travel along the desired heading with reasonable accuracy.

The navigation system provided the robot with an ability to achieve a task and operated at a high level of abstraction. The navigation system's ability to create successive movements was based on a mathematical calculation of the users desired travel trajectory, in relation to the robot's current position. This calculation was performed after every step taken by the robot, to adjust for the robot's geometry, and hence limited range of motion. The robot was able to achieve navigation by maintaining a history of its own position coordinates as it progressed through its movement, and was able to relate its coordinates to a destination provided as a target or goal.

5.2 The Region Analogy

The basis for the robotic navigation was derived from Spike's geometry. The 'Region Analogy' formed a basis for segregating the ground space of the robot into three segments that were based on the robot's current position on the ground. The robot's triangular footprint allowed travel in one of three directions that were each separated by 120°. At rest, the three legs placed on the ground created an equilateral triangle shape, and each of the sides of the triangle acted as a turning axis for the robot. The triangular footprint formed from the base of the robot, including tipping axes shown in Figure 5-1.

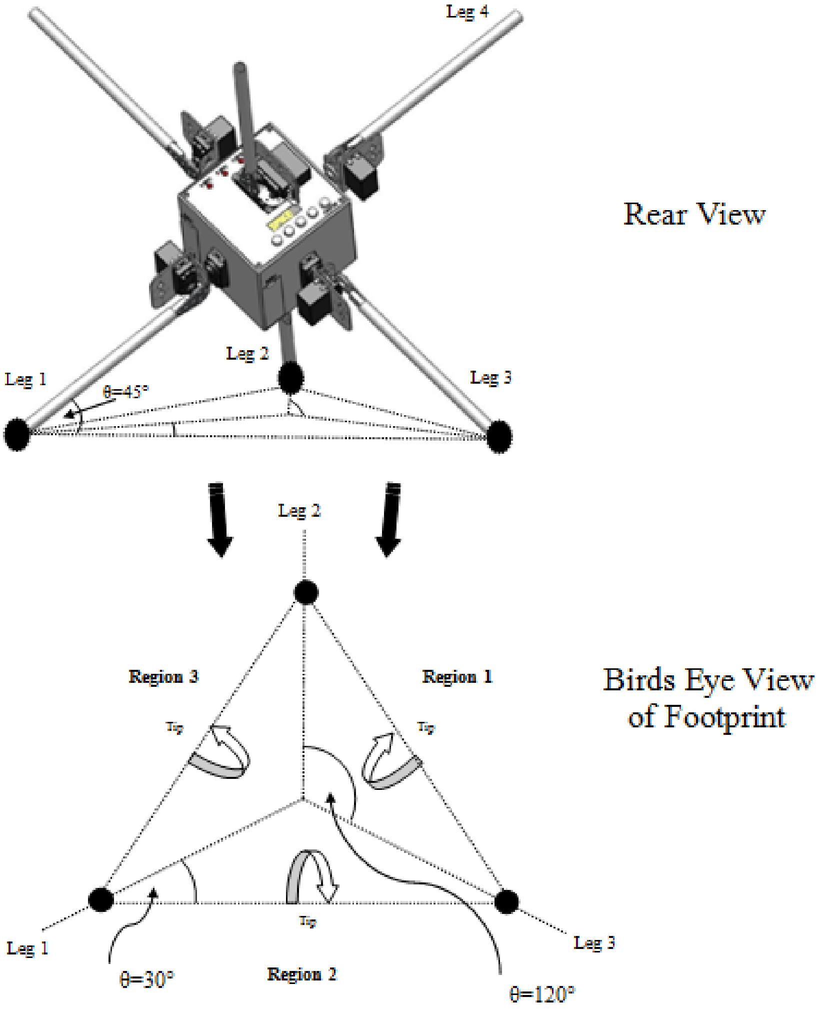


Figure 5-1: Description of Footprint

When examining the robot from a rear position, the triangular shape on the ground used either a single leg or two legs grounded at the rear of the robot creating two cases: 'Scenario 1' and 'Scenario 2' respectively. Figure 5-2 indicates these two scenarios. Each region is defined by a coloured bar, and represents a full 120° range from the footprint's centre. The two footprint scenarios are shown as follows, with their region definitions described in Table 5-1:

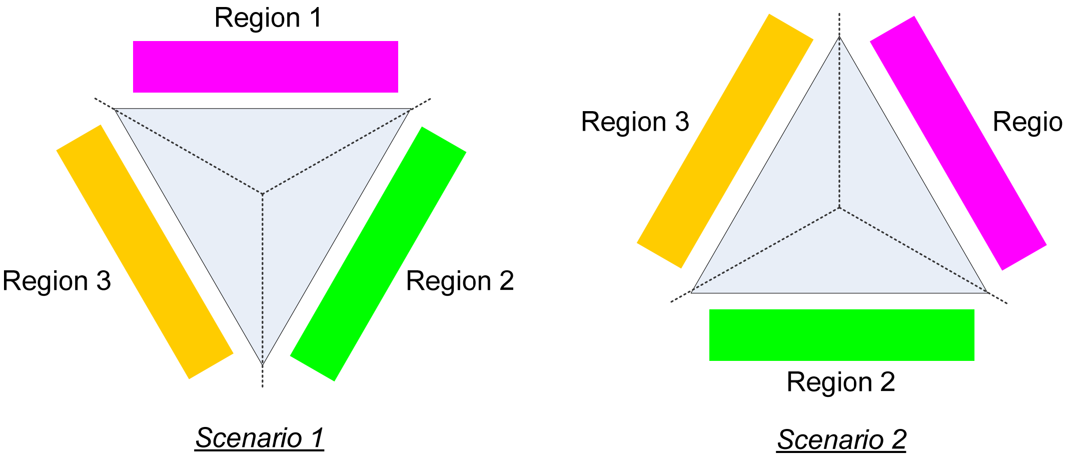


Figure 5-2: Indication of Regions from Footprint

	Scenario 1	Scenario 2
Region 1	Between 300° and 60°	Between 0° and 120°
Region 2	Between 60° and 180°	Between 120° and 240°
Region 3	Between 180° and 300°	Between 240° and 360°

Table 5-1: Definition of Regions

5.3 Navigation Strategy

The navigation process began with the user's placement of the robot on the ground in a predefined orientation. In this orientation, the centre of the robot was taken to be the origin, from which the robot would take steps and become displaced. The desired heading angle was 'dialled in' via a remote control device, and the desired angle was displayed on the robot's LCD screen as the user found their desired heading. After the user was satisfied with the selected heading, confirmation was performed by a sequence of button presses on the remote control, and then the navigation process began. The user was not required to make any further decisions of the robot's trajectory and the sequence of leg movements was completely controlled by software in the robot itself.

Before the navigation sequence began, the legs were defined with preset leg bearings, a numerical value that indicated the direction in which each leg was pointing, with respect to the robot's starting configuration. For example, the leg that pointed directly ahead of the robot was assigned a bearing of 0° as this leg was considered to be oriented north relative to the robot. As the robot has a total of six legs each leg bearing is a multiple of

60°. This is illustrated in Figure 5-3 with a bird's eye view of the robot in a 'Scenario 2' starting configuration. The legs that are grounded in the figure form the triangular footprint indicated by the dashed line.

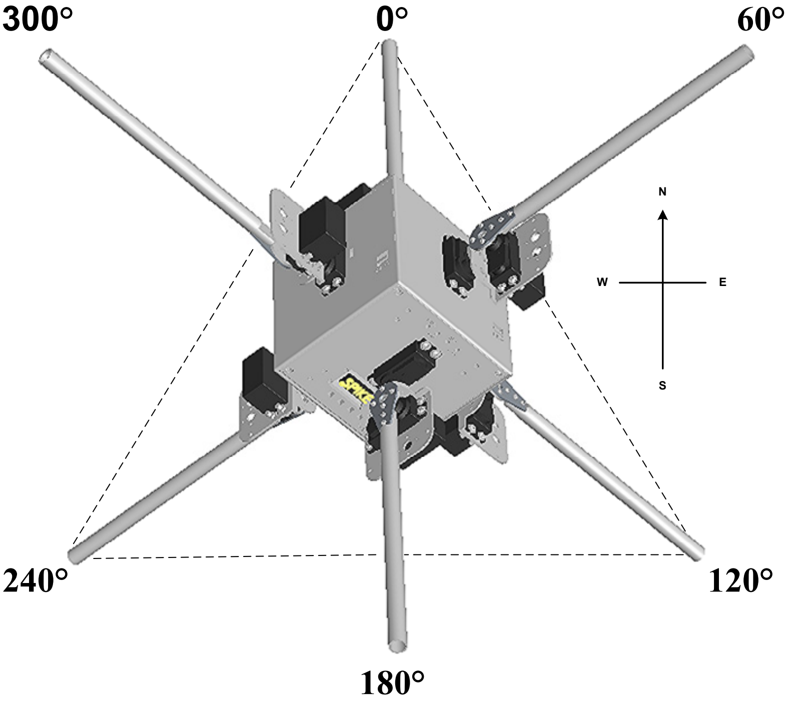


Figure 5-3: Birds Eye View Showing Leg Bearings

As the robot moves from one position to the next, the robot leg bearings change in a way that depends on the bearing of the driving leg. The new leg bearings however, are still all multiples of 60°. Using the orientation the robot was positioned in, combined with the driving leg's bearing and the region the robot was travelling toward, the new leg bearings could be calculated at the completion of each step the robot took. This is illustrated in Figure 5-4. Note the coloured leg ends are included to assist with determining the changing of leg positions and bearings made through the transition of the single step.

The robot received feedback from LDR light sensors as it progressed through each step. The combined usage of LDR light sensors with the calculated changes of the leg bearings, allowed the robot to calculate its dynamically changing orientation and direction in its environment after each step.

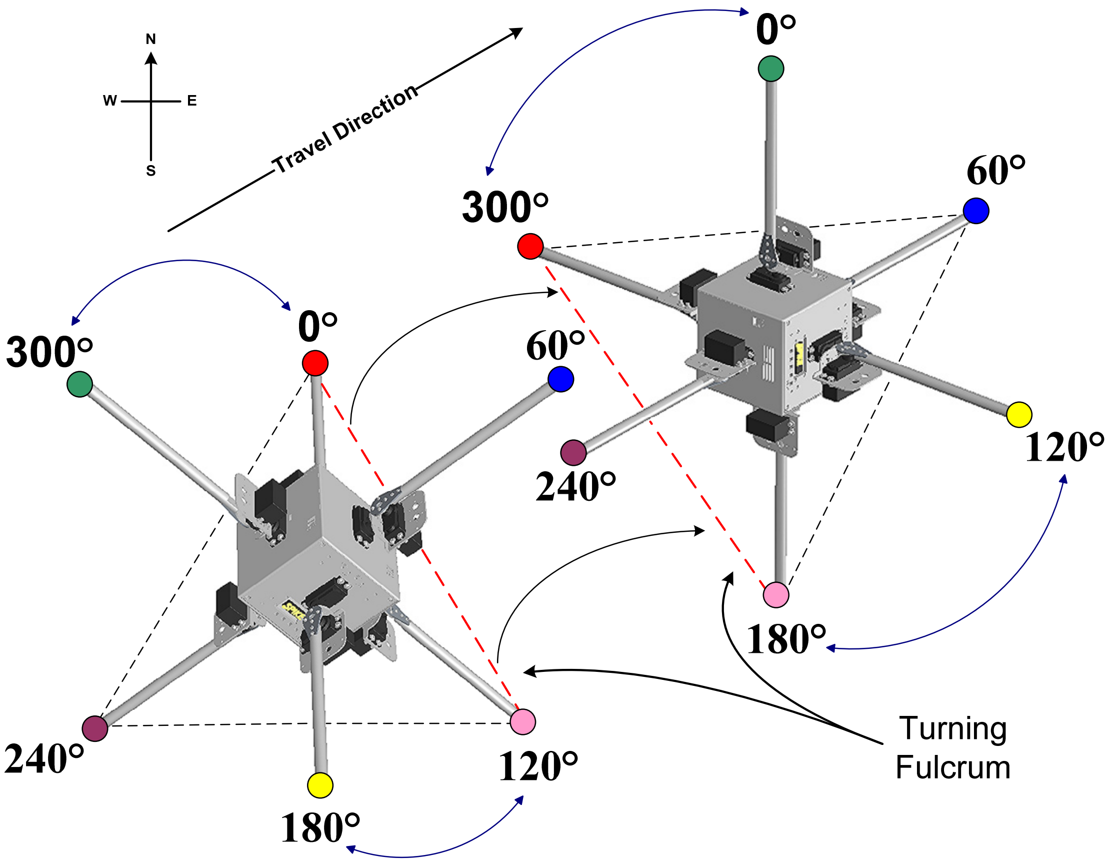


Figure 5-4: Birds Eye View Showing the Changing of Leg Bearings

The gait strategy for Spike is shown in Figure 5-5. Before commencing movement, it is assumed that the centre of mass is balanced evenly in the centre of the cube. To illustrate Spike's motion, consider the following example:

If the desired heading fell due north of the robot, the legs would need to arrange themselves in such a way as to allow the robot to fall in the direction nearest to due to north. In Figure 5-5, with the leg references used in Figure 5-2, this was performed by driving leg 1 directly toward the ground, in line towards the midpoint of the line drawn between legs 2 and 3. As the driving leg progressed through its range of motion, legs 2 and 3 acted as a fulcrum turning point. On tipping, the side of leg 1 rose up and toward the tipping point directly over the two remaining standing legs. After the robot's mass shifted past these two standing legs, the robot fell onto leg 4, and was hence in a new leg configuration and orientation.

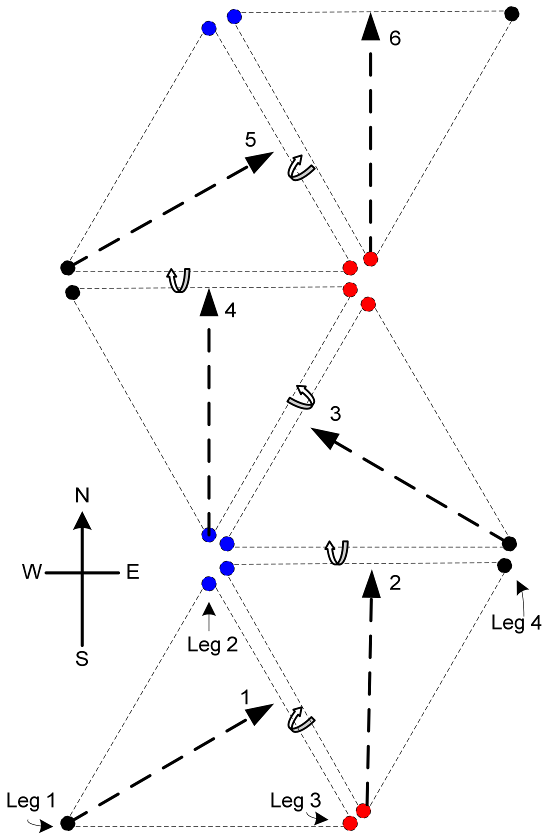


Figure 5-5: Gait Strategy

As the robot moved, its navigation system recalculated its current position to determine how the next step should change from the previous step, with the goal of travel in the desired direction of travel. This was necessary due to the limited number of directions the robot was able to tumble toward, meaning that the robot would occasionally need to take a sideways step, in order to achieve the required overall result.

The desired heading was used to calculate a trajectory for the robot to follow. As the robot moved, the navigation system calculated a target position for the robot to reach based on the robot's coordinate and the desired heading. The target position calculation used a defined separation distance of 1.5 times the robot's length to maintain a forward motion from its position. After each step, the target coordinate was shifted further along the heading trajectory to adjust for the robot's displacement from its starting position. The defined separation distance between the target and robot was chosen to reduce the temporary variations experienced by the robot away from the heading trajectory.

As the robot made successive tumbling steps, a record of its current location was kept and used with the original desired heading that the user instructed the robot to travel toward. The dynamic recalculation of the trajectory heading allowed the robot's motion to follow the user's desired heading. This change of robot position and change of imaginary 'target' position for a single step is illustrated in Figure 5-6.

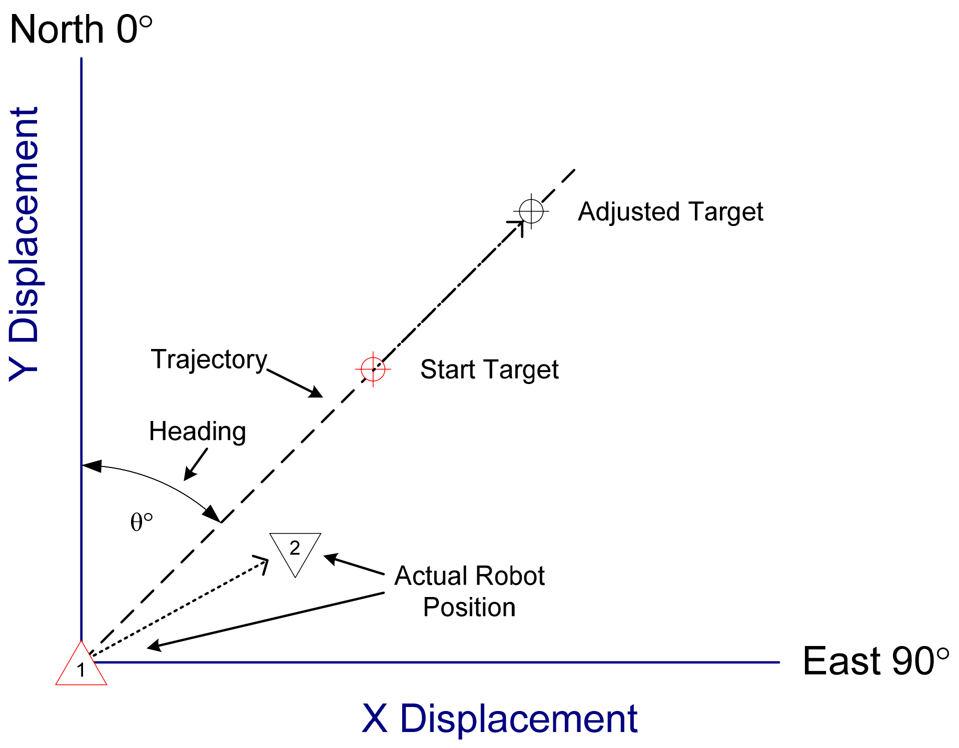


Figure 5-6: Illustration of Robot and Target coordinate Change

Due to the robot's footprint taking geometry from an equilateral triangle, the robot's position in its environment may be determined by calculation, based on the heading the robot was falling toward. For an example of travel at a 60° bearing from its current position, the new coordinates were determined by the robot's navigation system using the following equations:

$$xRobot = xRobotPrevious + 0.5 \times (RobotLength)$$

$$yRobot = yRobotPrevious + \left(RobotLength \times \frac{\sqrt{3}}{6} \right)$$

RobotLength is the length of the side of the equilateral triangle. These formulas are extracted from the triangular geometry shown pictorially in Figure 5-7.

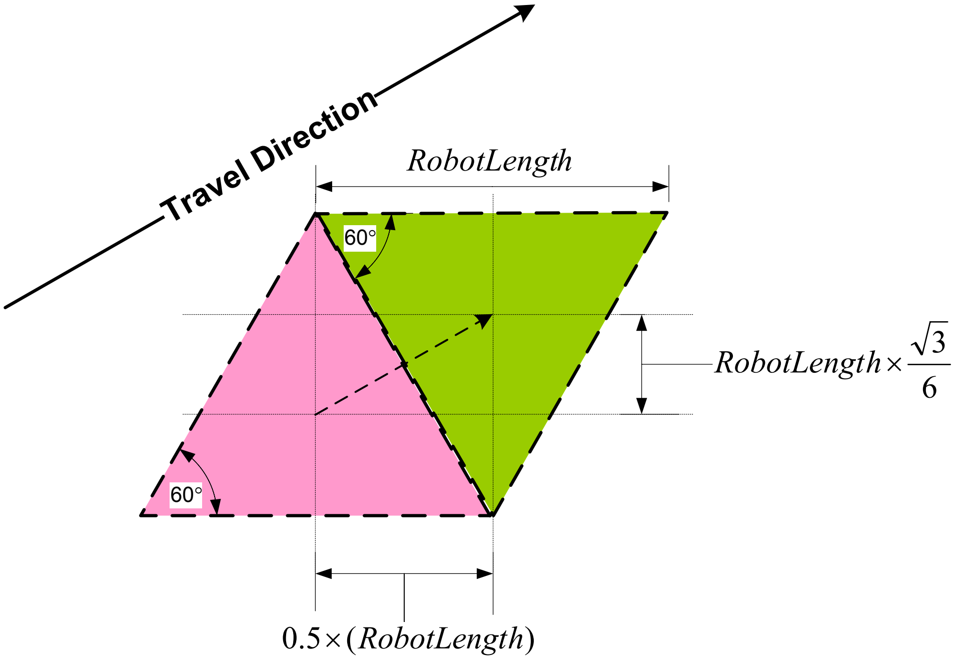

Travel Direction
$RobotLength$
60°
$RobotLength \times \frac{\sqrt{3}}{6}$
60°
$0.5 \times (RobotLength)$

Figure 5-7: Triangle Geometry for Coordinate Calculation

Following calculation of the robot's coordinate, the adjusted target position was determined. The target was defined to be a fixed distance from the robot along the required travel direction from the original starting position.

After each step the robot took, the target coordinate required adjustment to compensate for the robot's new position. The fixed distance to the target was added to the robot's distance from the starting position and the new target position was calculated as this distance along the required direction of travel.

The target coordinate was determined in software using:

$$DistanceFromStart = PresetDistance + \sqrt{xRobot^2 + yRobot^2}$$

$$xTarget = DistanceFromStart \times \sin(DesiredHeading)$$

$$yTarget = DistanceFromStart \times \cos(DesiredHeading)$$

From the new robot and target coordinates, the new trajectory for the robot was calculated as shown in Figure 5-8:

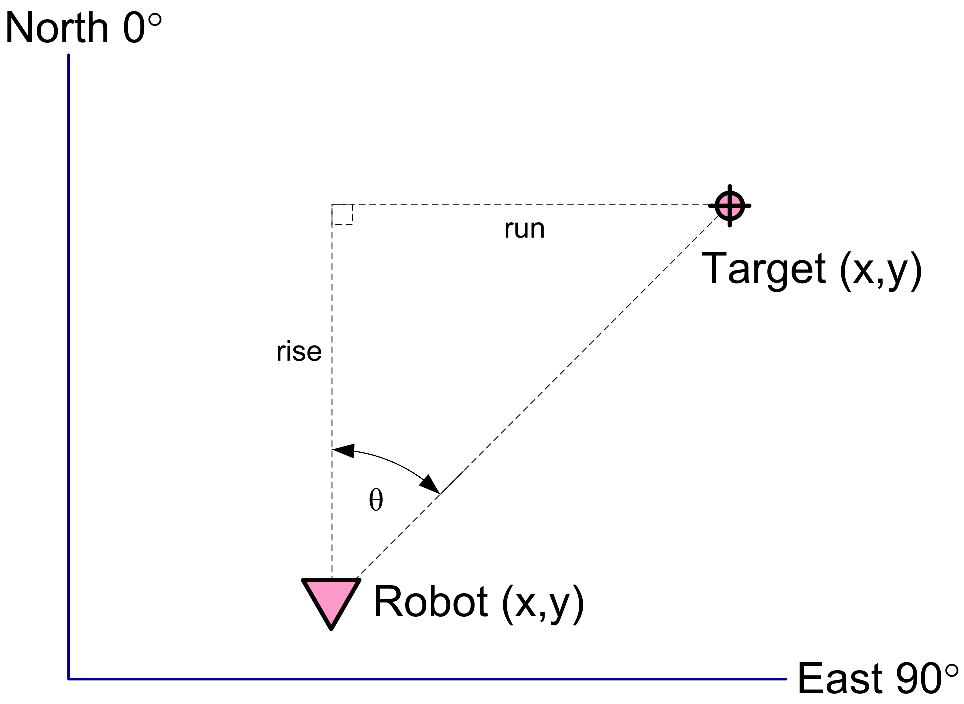


Figure 5-8: Calculation of New Trajectory

This new trajectory for the next step was determined using:

$$\theta = \tan^{-1}\left(\frac{run}{rise}\right)$$

The effectiveness of this navigation technique is explored in later chapters of the thesis. The navigation system developed on the robot provided a level of robotic control to be achieved. The ability of this system may be further developed to dynamically reconfigure the feet positions on the ground, to create alternative triangular footprints that would allow other variations of headings for the robot to travel toward, and hence provide an increased number of directions of travel.

6 SOFTWARE

6.1 Overview

The purpose of this chapter is to describe the notable software features in the programming of Spike the robot. The full program listing is attached to this thesis in a CD-ROM for reference, with necessary program comments. The task of the software used on Spike's systems was to perform decision making. The software was also responsible for allowing inter-processor communications to occur and to send control signals to legs and interpret readings given from sensors monitoring the robot's environment.

Firmware developed for two processors was used to provide the necessary processing and functionality for the robot. The software for Spike was developed entirely in ANSI C program language using the CodeVisionAVR integrated development environment. CodeVisionAVR is used throughout AUT University as a preferred development environment for programming Atmel AVR microcontrollers and was therefore readily available. Programming the target microcontrollers using AVRStudio4 and hardware JTAG ICE devices allowed the software to be developed quickly and accurately. The conceptual system design overview is shown in Figure 6-1.

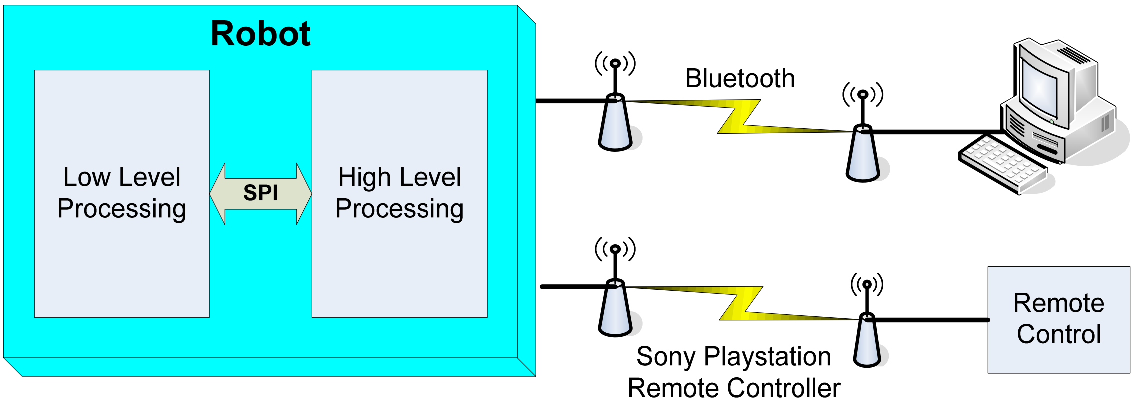


Figure 6-1: Spike Conceptual Overview

Inter-processor communication between the two Atmel AVRMega128 microcontrollers was achieved with the use of the microcontroller's built-in serial peripheral interface (SPI). This interface allowed a master/slave hierarchy of communication to be set up between the two microcontrollers where the low level processing motherboard was defined as the SPI slave, and the high level processing daughterboard was defined as the

SPI master. Transmitted data was stored into an array of programmed variables named ‘registers’, which was common to the two processors. Data transmission consisted of packets of data that included two bytes, address and data respectively. The transmitted address was a value between 0 and 40, and related to a specific ‘registers’ array element to be accessed. The data byte held a numerical value between 0 and 255. The address sent from the SPI master was used to manipulate a finite state machine that was programmed into the SPI slave. The register address sent from the SPI master determined the operation that was performed on the following data byte, to either read information contained at the array element or to write the data received into the array element.

Three programmed states were available in the state machine including an ADDRESS state, a READ state and a WRITE state. The ADDRESS state was defined as state 0, the initial state where the first byte of incoming data received would be recognised as a register address. The value of the address would determine the next state for the state machine, being either a READ or WRITE state. A total of 40 registers were declared in memory where 0-19 were programmed as writing addresses and the remaining 20 were programmed as read addresses. This provided an ability to send commands to the motherboard to write to a register address, and for the motherboard to prepare information from a register address that the daughterboard could subsequently read. The state machine was programmed into the SPI data transmission complete interrupt of the motherboard program. The interrupt service routine was triggered at the completion of a data byte transmission from the master device by the microcontroller’s hardware SPI sub-system.

If the transmitted address indicated a READ operation, the state machine was prepared for a read state. This was performed by initially storing the value of the address received, and then loading the contents of that register to the SPI data register. As the next byte of data was received, the value of the SPI data register was then transmitted back to the master device using the clock signals generated from the master device’s transmission. At the conclusion of the data transfer, the state machine was configured once again for an initial ADDRESS state in preparation for the next packet of data. The clocking of the SPI unit was necessary for the slave interface to transmit data to the master, as defined in the specification for the microcontroller. For a READ operation, the value of the data transmitted to the motherboard after the address byte was not relevant, but was required to provide the necessary clock pulsing. An example of a read instruction may be the interrogation of the current light readings from the LDR light sensors.

If the address byte indicated a WRITE operation request, the basic structure from the read operation was used with some additional commands to store the second byte of transmitted data. This was achieved using the address first sent to the state machine as a register address, and storing the second transmitted data byte into the defined register address. An example of data to be written may be a new position for a servo motor. The state machine operation is shown in Figure 6-2.

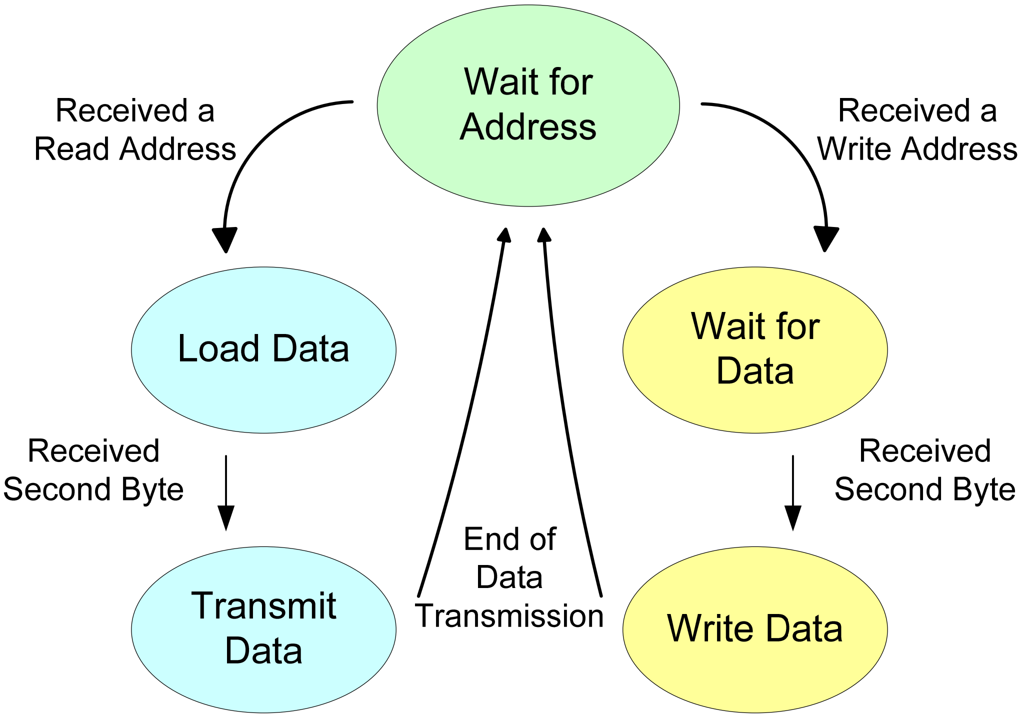


Figure 6-2: FSM of SPI Register Communication

The high and low level control method was maintained throughout the entire system design. The robot's control was separated into two levels of decision making. The decisions made about the navigation and robot orientation, were processed at a higher level of design to that of the individual pulsing of servo motors and analysis of light sensors. This method of design allowed the critical timing restraints and processing overhead posed by the software PWM method to be implemented successfully to drive the servo motors, and also facilitated modular control for the expansion of interchangeable control boards. By separation of the control, the robot was modular for its expandability and development possibilities.

The pin descriptors for the microcontrollers used in Spike are tabulated in Appendix A.

6.2 High Level Control – 'Daughterboard'

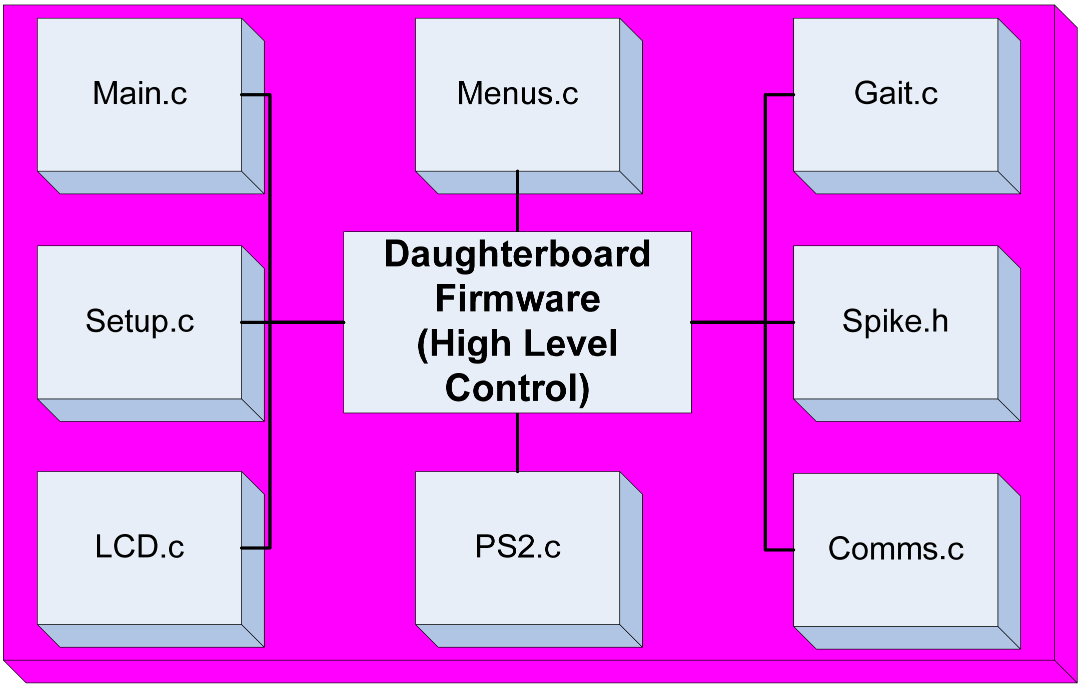


Figure 6-3: Daughterboard Project Files

The diagram shown in Figure 6-3 shows the project overview for the creation of the Daughterboard software. The program code spanned multiple source files. By splitting the program code into several individual files, the project had a modular structure that was scalable and easy to manipulate and navigate.

The software included the following files:

- Spike.h
  - Included all global project definitions including: header files, pin assignments, function declarations and variable declarations global to all source files.
- Main.c
  - Main program loop that included two primary function call selections. The selected function call was based on the top level hierarchy selection made by the user of either setup mode or remote control mode.

- Setup.c
    - Included the setup statements that defined the microcontroller's various control register bytes, and setup of the microcontroller settings such as: port directions and assignments, interrupt settings, timer settings, USART settings, internal pull-up resister settings etc.
- LCD.c
    - Included the various function calls for initialisation, communication and displaying characters on the LCD screen.
- Comms.c
    - Included the communication functions for bluetooth, SPI and provision for the zigbee module communication.
- PS2.c
    - Included the several functions required for the software bit bang serial communication to the remote controller. This also allowed the information gathered from the remote control to be used directly throughout the program code.
- Gait.c
    - This large source file included many functions and was responsible for all matters relating to the navigation and control of the walking abilities for the robot.
- Menus.c
    - This source file contained various functions that were called from the top level hierarchy menu system located in Main.c. The various configuration options in the setup menu selection were included, as well as calls for the robot remote control navigation.

The responsibility of the daughterboard programming was to assume the high level control of the entire robot system. Spike's high level control primarily consisted of locomotive decisions, analysis of locomotive sensory data, external communications and interfacing with the user. Considerable program code was developed to facilitate these functions and unite all systems to achieve the fully operational robot. Selections of the notable main features of the daughterboard program are discussed in this chapter.

6.2.1 Menu System

The robot used an on screen menu system, programmed into the daughterboard processor. This menu system allowed the user to interact and manipulate the features of the robot, and gather information from the robot's sensors. The 16x2 LCD screen was combined with four user buttons to provide the necessary I/O interaction between the robot and user. The LCD and button combination on the robot chassis is shown in Figure 6-4.

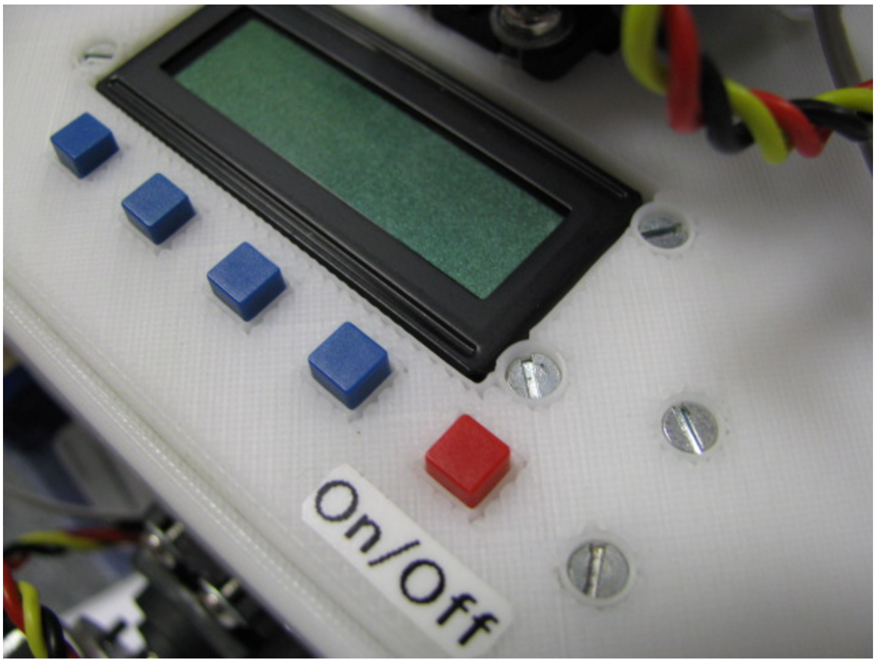


Figure 6-4: LCD Screen and Buttons

The menu system used on the LCD screen is outlined in Figure 6-5, Figure 6-6 and Figure 6-7.

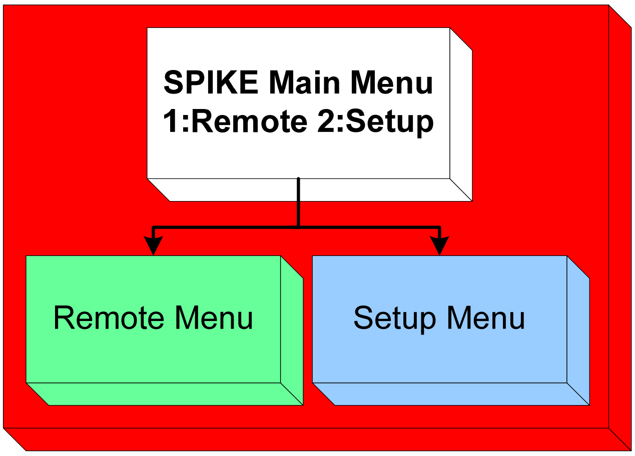


Figure 6-5: Top Level of Menu System

The four user buttons were used to navigate by scrolling, entering and exiting from the various robot menu options. The top level menu hierarchy presented the user with an option to select the operation mode for the robot. These two operation modes were; to either enter the setup menu where the status readings and configurations were manipulated, or to enter the remote control menu for robot navigation. The user was able to use the buttons mounted on the chassis to select either option that corresponded with the selection choices presented on the LCD screen. The setup menu selection options are shown in Figure 6-6.

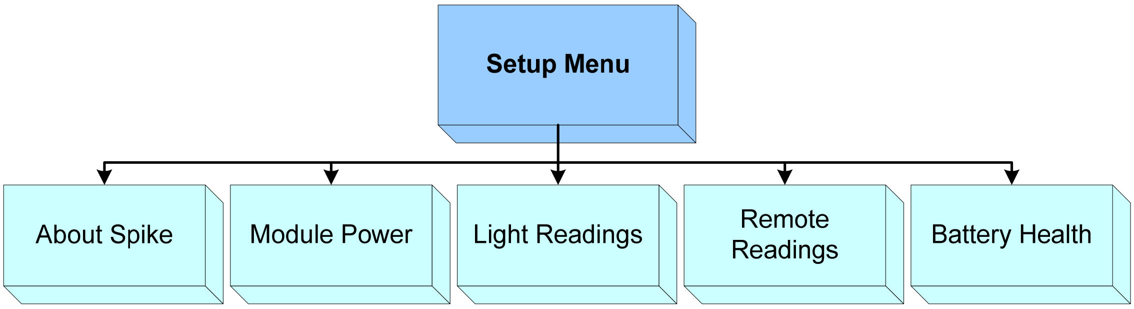


Figure 6-6: Setup Menu

The setup menu included five sub menus as shown in Figure 6-6. The navigation through the menu was provided with left and right arrows for scrolling, and an OK and Back selection option. The user was able to navigate within these sub-menus by using the navigation buttons. A brief explanation of the sub menus follows:

- Battery health
  - Displays the current voltage level of all four batteries.

- Remote Readings
  - The button presses on the remote control are displayed within this menu for clarification of the remote control communication and operation.
- Light Readings
  - The light readings taken from the LDR light sensors of all six legs are displayed on the LCD screen.
- Module Power
  - The device power for the bluetooth and zigbee modules may be toggled within this menu.
- About Spike
  - Details of the author and software version.

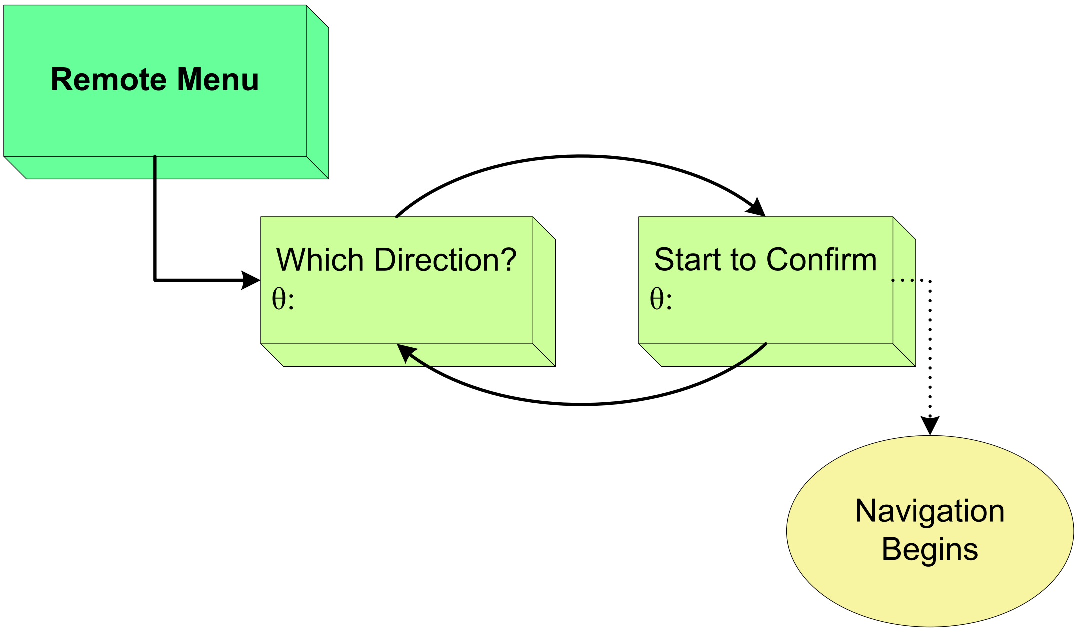


Figure 6-7: Remote Menu

Figure 6-7 shows the operation of the remote control menu. The screen prompt displayed ‘Which Direction?’ and ‘Start to Confirm’ at a half second rate, as the user directed the analogue remote control joystick to the desired heading. While the display toggled between display messages, the current angle of the joystick was actively displayed on the bottom section of the screen area. If the robot failed to detect the presence of the remote control, the robot displayed an appropriate error message on the LCD screen to alert the user to the loss of communication. Following establishment of the communication, the remote control menu was then displayed correctly. To confirm the desired heading, the user combined pressing of the remote control's 'Start' button whilst simultaneously holding the joystick toward the desired direction. Using this method, the robot was able to capture the desired angle, and begin the navigation process toward the desired heading. Any further direction changes of the remote control joystick following the confirmation of selection were discarded.

Toggling between menu screen messages was performed by an accumulation of an interrupt counter. This allowed the active communication between the remote controller and the LCD display to occur simultaneously with the remaining processing time of the microcontroller.

- Which Direction? / Start to Confirm?
    - Toggled between the two display messages. The program continued with the navigation functions after the user confirmed the desired heading with the simultaneous pressing of the remote control's start button with the joystick directed toward the desired heading.

The program structure used on the robot's menu design provided a level of flexibility to facilitate the integration of subsequent iterations of the robot's features as they were developed.

6.2.2 Gait

The development of Spike's gait was the primary function of the entire programming operation. Software control to generate the robot's motion created the foundation from which results and analysis of the entire system could be formulated. To allow the gait to be developed, many sub-systems required implementation, e.g. inter-processor communication, such that the high level gait operations could be executed. The structure for the gait's operation is shown in Figure 6-8. The logical flow of the gait operation consisted initially of gathering information used to determine the motion task, followed by execution of the motion from the information gathered and calculations performed.

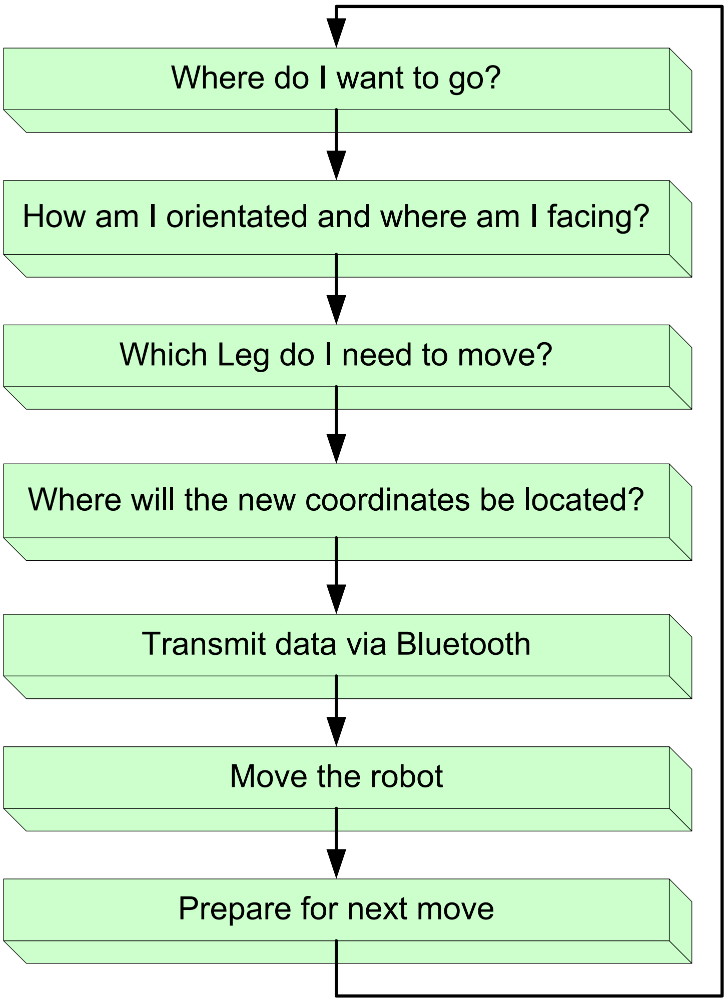
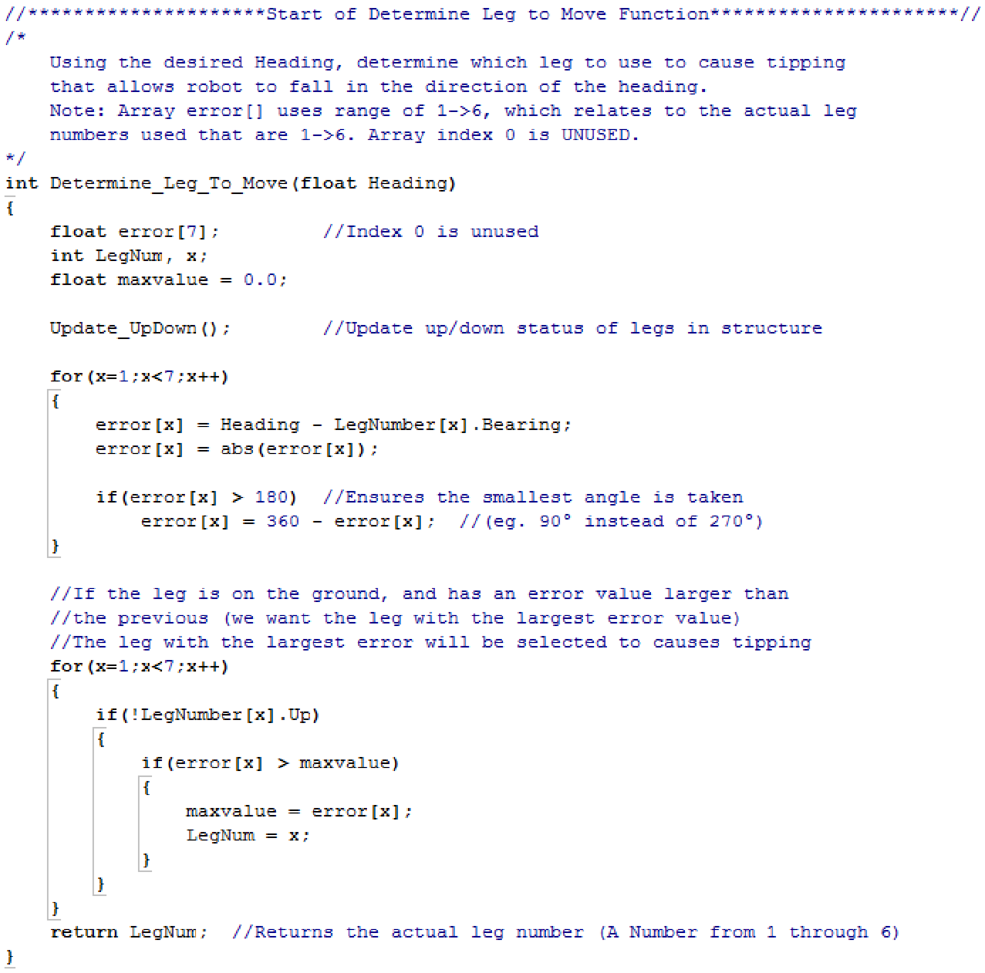

Where do I want to go?
How am I orientated and where am I facing?
Which Leg do I need to move?
Where will the new coordinates be located?
Transmit data via Bluetooth
Move the robot
Prepare for next move

Figure 6-8: Flow Diagram of Gait Instructions

The robot's motion ability was facilitated with the use of: navigation, wireless communication, calculation of the required servo motor positions, reading of sensors, inter-processor communication and interfacing with the user. These features aided achieving a successful overall motion. The programming of the gait also incorporated an error checking ability to respond to potential errors generated from calculation, and to ensure correct sensor measurement during the navigation process. Errors were displayed on the robot's LCD screen to prompt the user for acknowledgement to resolve any error. The Gait source file assumed the overall system responsibility for the entire robot motion.

Included in the gait functions were calculations to determine the correct leg to move on each step the robot took. This function of the gait required the robot to consider its orientation, the desired trajectory heading and the bearings of each of the robot's legs. The selection of the correct leg to move was critical to the entire navigation system, and experienced dynamic change as the robot moved in its environment. The C code function is included in Figure 6-9 to show the operation of this critical function.



```
//*********************Start of Determine Leg to Move Function*********************//
/*
    Using the desired Heading, determine which leg to use to cause tipping
    that allows robot to fall in the direction of the heading.
    Note: Array error[] uses range of 1->6, which relates to the actual leg
    numbers used that are 1->6. Array index 0 is UNUSED.
*/
int Determine_Leg_To_Move(float Heading)
{
    float error[7];         //Index 0 is unused
    int LegNum, x;
    float maxvalue = 0.0;

    Update_UpDown();        //Update up/down status of legs in structure

    for(x=1;x<7;x++)
    {
        error[x] = Heading - LegNumber[x].Bearing;
        error[x] = abs(error[x]);

        if(error[x] > 180)  //Ensures the smallest angle is taken
            error[x] = 360 - error[x];  //(eg. 90° instead of 270°)
    }

    //If the leg is on the ground, and has an error value larger than
    //the previous (we want the leg with the largest error value)
    //The leg with the largest error will be selected to causes tipping
    for(x=1;x<7;x++)
    {
        if(!LegNumber[x].Up)
        {
            if(error[x] > maxvalue)
            {
                maxvalue = error[x];
                LegNum = x;
            }
        }
    }
    return LegNum;  //Returns the actual leg number (A Number from 1 through 6)
}
```

Figure 6-9: Code Example showing Determination of Correct Leg to Move

To determine the leg to move, the function firstly examined all six LDR light sensors of the legs to determine the light level's experienced by the robot. The three legs that experienced the lowest light levels were assumed to be located on the ground surface. After the orientation of the robot had been determined with the light sensors, the user's desired heading was compared to the bearing of each of the legs. The outcome of the comparison presented a numerical value 'error', representative of the leg's location and direction to the desired heading trajectory.

The leg experiencing the greatest error was used to drive the robot to form the robot's next step. A greater value of error indicated that the leg was directed away from the desired heading, and thus was best fit for creating the motion based on the robot's triangular geometry. Using knowledge of the three legs placed on the ground, the function was able to formulate the greatest error value, and the chosen leg was consequently returned as a leg number, to the calling function for further numerical manipulation.

To position the motors correctly, a large lookup table was programmed into the processor’s memory. The lookup table defined the exact position for each of the twelve servo motors for the entire robot to reach, to correctly position each leg regardless of the robot’s orientation. The lookup table was defined as a signed three dimensional array that held all eight possible robot orientations, including three direction options for each orientation, with each option holding twelve motor displacement values relative to the starting position for every motor. This method allowed the orientation of the robot to first be determined using the LDR light sensors, which affected the first dimension of the lookup table. The second dimension was determined by calculation of the individual leg required to cause tipping toward the desired heading and consequently, the motor positions were determined.

After the tipping leg was determined, the function’s return value of the proposed driving leg was converted into an index value of 0, 1 or 2, relating to one of the three direction possibilities most appropriate for motion. The twelve motor values stored in the third dimension of the array were then sent to the motherboard to operate the motors.

Each element of the lookup table was defined by a meaningful word or abbreviation such as ‘Cat’ (Catch), ‘Tip’ (Tipping) or ‘Rot’ (Rotate). For example, if the two servo motors were required to be positioned to allow the leg to catch the robot, their displacement values were set to have a base rotation defined by ‘Rot’ and joint actuation defined by ‘Cat’. All of the table’s 288 elements were defined in this manner allowing all values to be quickly modified in the project’s header file if required.

After retrieving the correct sequence of motor displacement values from the lookup table, the values were added to the default motor starting position of 90° which formed an exact angular position for each servo motor to reach. For example, if the lookup table value for ‘Rot’ was defined as 45, and a motor required adjustment by ‘-Rot’, the new servo motor position would become 45°. Similarly, if the same motor were to adjust by ‘Rot’, the new position would become 135°. This angular position was transmitted to the motherboard using the SPI interface. The motherboard then assumed responsibility for converting the set angular position into a servo motor pulse width modulation signal, used to control each motor position through a number of interrupt driven steps, such that the desired position was obtained.

6.2.3 Wireless Communications

6.2.3.1 Remote Control

The Sony Playstation® 2 styled remote control communicated with the onboard microcontroller using an SPI bit banging technique. Communication with the remote control's interface had been specifically designed for use with a Sony Playstation® and not with this robotic application and therefore, investigation into a method of communicating data was required. Through a series of pulse timing tests with the remote control's interface, and the subsequent analysis of communication data, a specific method, in program code, to achieve the required pulse timing was discovered, and the non-standard bit banging technique was constructed. Through communication with the remote control using this method, identification of button status and analogue joystick position could be accurately determined. Development of this interface was performed with assistance from various internet forums and discussion with peers.

In order to obtain data from the remote control, the design of the controlling functions in software were crafted according to the preconfigured command structure shown in Figure 6-10.

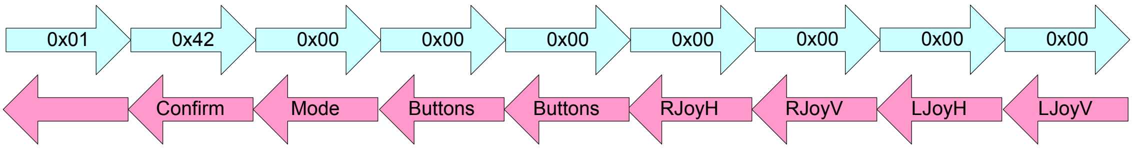
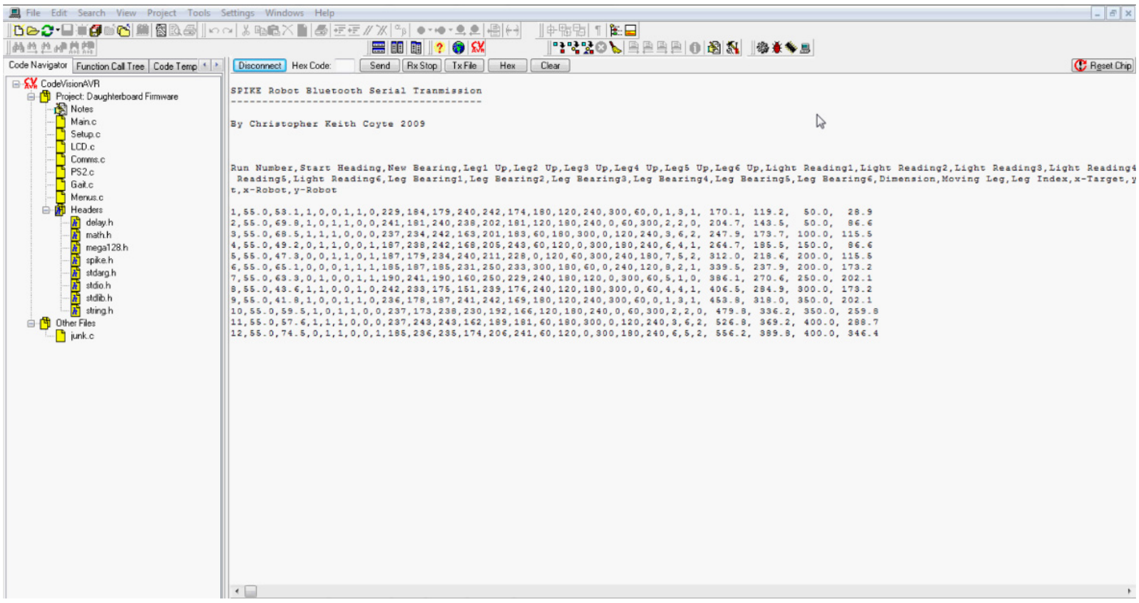


Figure 6-10: Remote Control Command Structure

For two-way communication to occur, the robot was required to clock out transmission data that consequently allowed data to be received and stored. The robot initiated communication by transmitting a 0x01 on its CMD (command) line while the CLK (clock line) was also toggled. The second transmission of 0x42 from the robot, allowed the remote control to respond with a confirmation byte of its running operation. The running operation of the remote control used in this development was of 'Analogue Controller in Red Mode' providing a 0x73 response from the controller. Following the 'running operation' response, a 'get ready' byte was transmitted, and the subsequent transmissions contained relevant data of the button's status and joystick positions. Further detail of the command protocol may be found online [88].

6.2.3.2 Bluetooth

Data prepared for transmission to a PC was sent via the microcontrollers built in USART (universal synchronous/asynchronous receiver/transmitter) interface, where the bluetooth module was attached. The bluetooth module acted as a serial transmit pipe for the robot at a rate of 38.4kbps, allowing numerical data from the robot to be extrinsically analysed on a desktop PC. The data from the robot was captured via a terminal program running on a nearby PC, and the data was saved to a file. The file was later used in a spreadsheet program (Microsoft Excel) to graph and analyse the data. A screenshot of the data acquisition process on the PC is shown in Figure 6-11.



```
SPIKE Robot Bluetooth Serial Tranmission
----------------------------------------

By Christopher Keith Coyte 2009


Run Number,Start Heading,New Bearing,Leg1 Up,Leg2 Up,Leg3 Up,Leg4 Up,Leg5 Up,Leg6 Up,Light Reading1,Light Reading2,Light Reading3,Light Reading4
 Reading5,Light Reading6,Leg Bearing1,Leg Bearing2,Leg Bearing3,Leg Bearing4,Leg Bearing5,Leg Bearing6,Dimension,Moving Leg,Leg Index,x-Target,y
t,x-Robot,y-Robot

1,55.0,53.1,1,0,0,1,1,0,229,184,179,240,242,174,180,120,240,300,60,0,1,3,1,  170.1,  119.2,   50.0,   28.9
2,55.0,69.8,1,0,1,1,0,0,241,181,240,238,202,181,120,180,240,0,60,300,2,2,0,  204.7,  143.5,   50.0,   86.6
3,55.0,68.5,1,1,1,0,0,0,237,234,242,163,201,183,60,180,300,0,120,240,3,6,2,  247.9,  173.7,  100.0,  115.5
4,55.0,49.2,0,1,1,0,0,1,187,238,242,168,205,243,60,120,0,300,180,240,6,4,1,  264.7,  185.5,  150.0,   86.6
5,55.0,47.3,0,0,1,1,0,1,187,179,234,240,211,228,0,120,60,300,240,180,7,5,2,  312.0,  218.6,  200.0,  115.5
6,55.0,65.1,0,0,0,1,1,1,185,187,185,231,250,233,300,180,60,0,240,120,8,2,1,  339.5,  237.9,  200.0,  173.2
7,55.0,63.3,0,1,0,0,1,1,190,241,190,160,250,229,240,180,120,0,300,60,5,1,0,  386.1,  270.6,  250.0,  202.1
8,55.0,43.6,1,1,0,0,1,0,242,233,175,151,239,176,240,120,180,300,0,60,4,4,1,  406.5,  284.9,  300.0,  173.2
9,55.0,41.8,1,0,0,1,1,0,236,178,187,241,242,169,180,120,240,300,60,0,1,3,1,  453.8,  318.0,  350.0,  202.1
10,55.0,59.5,1,0,1,1,0,0,237,173,238,230,192,166,120,180,240,0,60,300,2,2,0,  479.8,  336.2,  350.0,  259.8
11,55.0,57.6,1,1,1,0,0,0,237,243,243,162,189,181,60,180,300,0,120,240,3,6,2,  526.8,  369.2,  400.0,  288.7
12,55.0,74.5,0,1,1,0,0,1,185,236,235,174,206,241,60,120,0,300,180,240,6,5,2,  556.2,  389.8,  400.0,  346.4
```

Figure 6-11: Screenshot of Data Acquisition Process

Data transmitted via the bluetooth module included: step number, desired heading, robot determined heading, leg up or down state for all legs, light readings of all legs, leg bearings for all legs, robot orientation, tipping leg number, tipping leg index position, x & y coordinate of target, x & y coordinate of robot. This data was transmitted from the robot before each step was taken.

6.2.3.3 Zigbee

A zigbee wireless transmission module was included within the robot chassis whose functionalities may be implemented at a later stage. This device included specialised hardware that allow the integrated circuits to measure wireless signal strength between itself, and reference nodes placed within a short proximity of the module. By combining several reference nodes at fixed locations with defined separation distances, the wireless receiver within the robot may be configured to detect its position with reference to these nodes. This system may be beneficial where the actual physical displacement of the robot must be determined from a source external to the robot. A nearby PC monitoring the operation of the robot may use the coordinates determined by the zigbee module for evaluation of the effectiveness of various controllers for the robot.

The zigbee module interfaces to the microcontroller using a USART interface as for the bluetooth module, and is able to determine the coordinates of the robot's location with respect to the environment reference nodes. The coordinates received by the microcontroller may then be packaged for bluetooth transmission to the monitoring PC. This would allow comparison of the mathematically determined position to be compared to the actual physical position, to enable adjustment for real world factors, such as leg slippage. The zigbee module is included within the robot, but has been left un-programmed, where future software development will facilitate its use.

The zigbee module used in this application is the Chipcon (by Texas Instruments) CC2431 designed as the first system on a chip with a hardware location engine on the low-power zigbee interface. The motivation behind the development of this device was to extend the abilities of wireless sensor networking applications. The device is programmable in ANSI C, and the manufacturer specification for the position accuracy is stated to be 25cm, in its datasheet [89].

6.3 Low Level Control – 'Motherboard'

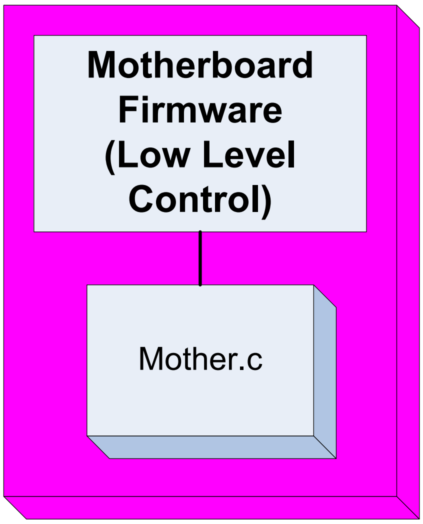


Figure 6-12: Motherboard Project Files

The diagram shown in Figure 6-12 shows the project files created in the development of the Motherboard firmware. The single ANSI C file 'Mother.c' held the necessary program code to enable the motherboard to behave as the SPI slave in the multiprocessor communication system, and also interpret sensory information and correctly pulse the servo motors based on the daughterboard's instructions. The motherboard was also responsible for controlling the power supplies to the remaining robotic systems.

6.3.1 Power Control

To achieve control of the robot's power systems, all individual components of the robot's circuitry required individual switching by the microcontroller. The microcontroller itself entered a low power sleep mode when the robotic system had been instructed to power down. This was achieved with the use of the power save features of the microcontroller. The 'power-down' mode enabled the microcontroller to be held in a low power state until the user toggled the soft 'power button' to generate an interrupt for the microcontroller. With an interrupt generated on the microcontroller's external interrupt line, the robot would restart its operation from where it was prior to entering the low power state.

Prior to the activation of the microcontroller power-down function, all devices whose power was affected by the motherboard's power control were shut down. By doing so, all external devices were prevented from drawing current from the general I/O pins of the motherboard microcontroller, and hence the microcontroller's current consumption was reduced to approximately 25µA. This current consumption was deemed acceptable for storage of the robot when powered off.

In software, this functionality was programmed with correct usage of register setup and built-in sleep instruction calls to configure the device to enter or exit low power states. Upon reactivation of the microcontroller, the program counter resumed from the line of code following the power-down instruction. After the microcontroller was brought online, the individual devices of the robot were reinitialised to facilitate correct operation. This involved enabling the power supplies for all devices including the daughterboard processor, and allowing the inter-processor communications to be re-established.

The motherboard also held responsibility to gather measurements of the battery voltages, LDR light sensor readings and temperature sensor reading that would later influence the operation of the robot. As previously discussed, the LDR light sensors were used as part of the robot's navigation system for determining the robot's orientation. Battery readings were also measured and monitored for safe and continued robot operation.

The battery voltage readings were displayed to the user from within the robot's setup menu system, and were values based on the analogue to digital measurements made by the microcontroller. The voltage was determined from an 8 bit ADC measurement using an equation drawn by recording ADC readings at known voltage values. This was performed using an external DC power supply of measured voltage and applying the source voltage through a voltage divider circuit on the motherboard that allowed suitable interfacing to the microcontroller's ADC input. The ADC readings were noted at various input voltages, and a linear relationship was drawn from the measurements. The gradient of the linear slope was found to be 0.0368, and hence a formula was drawn. The formula drawn from the measured findings is described below, and allows the 8 bit ADC reading of the battery voltage to be converted into a numerical measurement of the actual battery voltage.

$$BatteryVoltage = \frac{(Value_{ADC} \times 3.68)}{100}$$

Using this equation, the voltage of each battery was determined.

6.3.2 Servo Pulsing

The servo motor position control was implemented using software driven pulse width modulation (PWM). This allowed a physically small and inexpensive microcontroller to be used for the PWM generation required for the robotic control. As the microcontroller was required to drive 12 servo motors, which exceeded the number of onboard hardware PWM channels, software bit-banging was used to create the PWM signals. This software pulsing technique allowed simultaneous pulsing of the twelve servo motors for this robotic application. Twelve general I/O pins of the microcontroller were connected directly to the servo motor control lines that delivered the necessary pulse signalling for position control.

To adjust the position of each servo motor, the daughterboard sent the motherboard the desired destination position for each servo motor. This position was passed to the motherboard using the SPI inter-processor communication previously described in section 6.1. The communication consisted of a packet of 2 data bytes including the servo motor number as a register address byte and the desired angle in degrees for the motor to reach as a data byte. The acceptable range of motor position angles lay between 0° and 180°, where 90° represented a default position for each motor. The default (rest) position was where all legs became perpendicular to the face of the cube chassis and their respective side (90°).

By using a software pulsing method, delivery of the control pulses was maintained continuously to the servo motors, which consequently allowed each servo motor to maintain the desired position until the pulse width was modified for the servomotor to go to a new position.

Prior to the configuration of the motor pulse timing, the motion of each servo motor was calibrated, to ensure the response from all of the motors was uniform. This was performed by sending a set of pulses to each motor, and monitoring the destination reached by the output shaft. A measurement box (Figure 6-13) was constructed where the motor rotated an extension arm around an arc. The position of the arm was measured at three specific points 0°, 90° and 180°, to ensure that the rotation and final arm position corresponded to a given position in degrees from the control software.

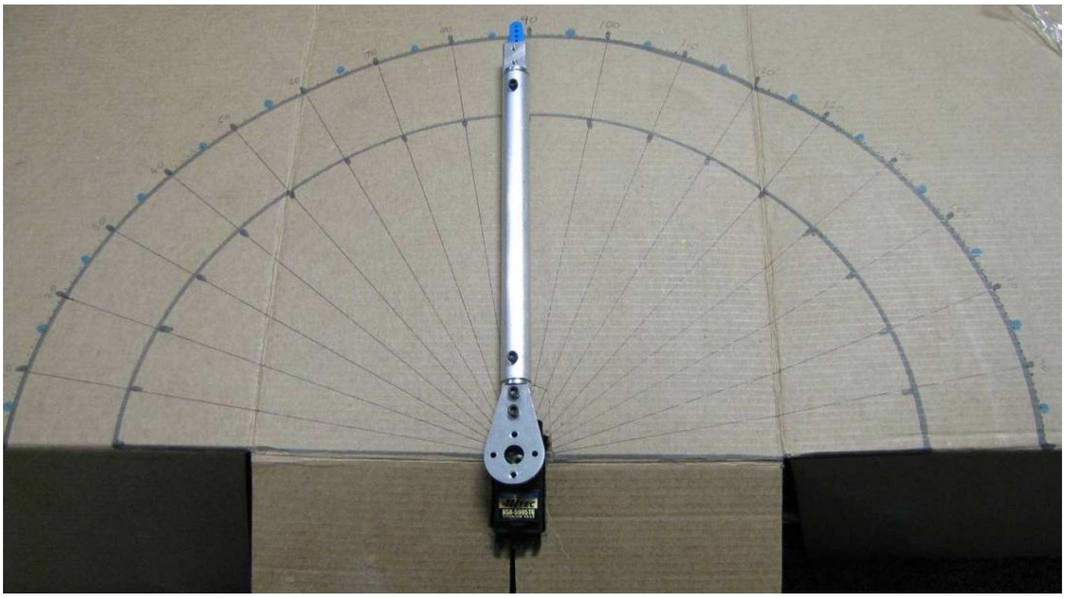


Figure 6-13: Servo Motor Calibration Box

Initially, the calibration process was performed by instructing all motors to reach the default 90° middle position. This position was deemed the most important, as the method of creating the entire robot motion was based on the symmetry of the robot with the legs directed at this position. Through testing, a large variation in obtained shaft positions was found and consequently the necessary adjustments were recorded and made. Calibration of the servo motors was performed both physically and in software, if the individual motor did not reach the required 90° position. The physical calibration involved removal and replacement of the servo motor horn mounted on top of the motor shaft, where adjustment was physically limited to increments of 15° by the manufacturer's machined motor spline. The software adjustment involved modification of the number of interrupts used to create the servo motor pulse width. The software adjustment allowed the accuracy of the motor shaft position to be modified to be within ±2.5° for the HSR-5995TG servo motors, and ±1.1° for the HSR-5990TG motors. This difference was due to their individual pulse range requirements that are discussed later in this section. With all motors reaching the default 90° middle position, the testing subsequently extended to ensure both 0° and 180° were correctly obtained. Table 6-1 shows the calibration adjustments used in software to correct position errors of the servo motors.

Table of Software Calibrations		
	Adjustment	Angular Offset
Servo 1	-2	2.2°
Servo 2	0	0°
Servo 3	-1	1.1°
Servo 4	0	0°
Servo 5	-2	2.2°
Servo 6	0	0°
Servo 7	-2	2.2°
Servo 8	0	0°
Servo 9	-1	1.1°
Servo 10	-2	5°
Servo 11	-2	2.2°
Servo 12	-2	5°

Table 6-1: Table of Software Calibrations

With all twelve servo motors calibrated for uniformity, and the individual software adjustments made, the construction of the software pulse timing for the robot began. Two different pulse ranges were used to match the timing requirements for the two different types of servo motors used, as per the manufacturer specification. An illustrative pulse range that corresponds with motor shaft position is presented in Figure

6-14 with the actual pulse width ranges detailed in Table 6-2 as per the manufacturer specification.

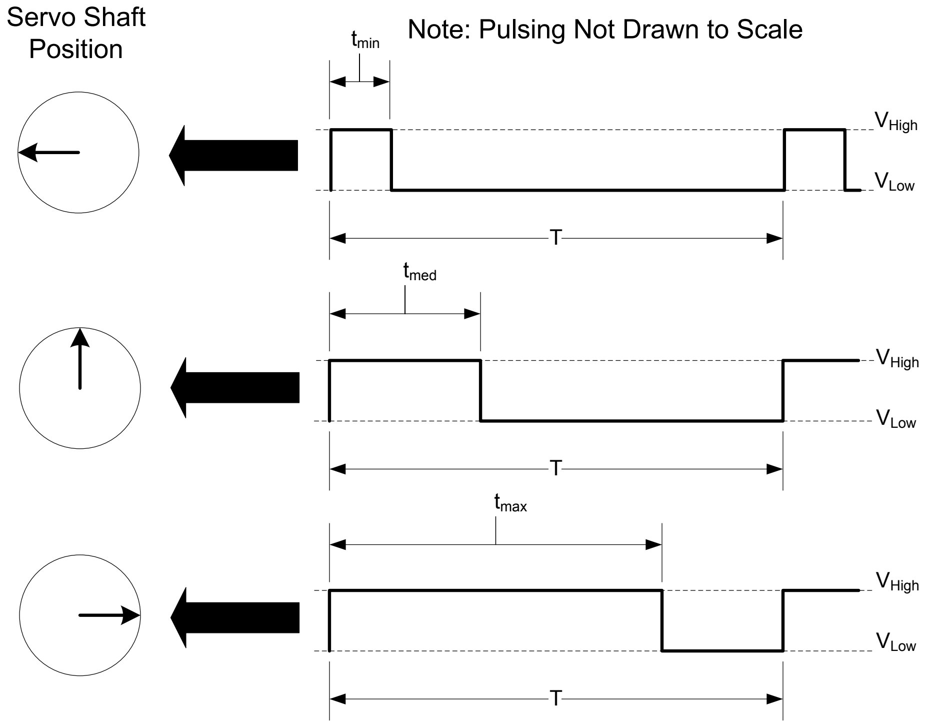


Figure 6-14: Servo Motor Control Pulsing

	HSR-5995TG	HSR-5990TG
t_{min}	1100μs	600μs
t_{med}	1500μs	1500μs
t_{max}	1900μs	2400μs
T	20ms	20ms

Table 6-2: Servo Motor Manufacturer Specification for Pulse Range

Bit-banging software was developed that would produce the required PWM timing pulses on each of the 12 PWM control pins of the microcontroller. The length of the pulse width was the most important part of the PWM waveform as this was what dictated the motor shaft position. In order to be as accurate as possible, an onboard timer driven from a 12MHz quartz crystal oscillator was used to generate regular interrupts that were used to trigger the bit-banging routine. Software pulsing of a single motor was achieved by driving the microcontroller output pin to a high state for a variable length of time before pulling the line low. This process was repeated every 20ms to conform to the manufacturer specification for the pulsing of the motors.

The precision of the pulse width was determined by how often the interrupt occurred, with a higher interrupt rate providing a higher motor shaft precision. The frequency of the interrupt was limited by the amount of processing that was required within one interrupt cycle, i.e. the software needed to check at the state of each of the 12 bit-banged PWM channels to determine if any action was required. The microcontroller was also responsible for other tasks such as reading the ADC's and communicating to the daughterboard processor, which all required processing time to complete. To provide a balance between servo motor position accuracy and spare available processing time for other tasks, careful consideration was paid to the overall timing requirements. To determine the processing overhead imposed by using software bit-banging as described above, the interrupt duration was measured using an external oscilloscope. The interrupt was completed after approximately 14μs, and consequently it was decided to use an interrupt period of 22μs in order provide 8μs of spare processing time to execute other operations. By using a 22μs interrupt period, a servo accuracy of 4.95° was achieved for the HSR-5995TG servos and 2.2° for the HSR-5990TG servos. This accuracy was considered acceptable for the motion of the robot. The overhead posed by the software bit-banging technique was therefore found to be 63%.

To achieve a 22μs interrupt, the 12MHz quartz crystal was pre-scaled by a factor of eight, thus the timer interrupt counter incremented at a 1.5MHz rate (667ns period). The interrupt routine was triggered when the timer/counter matched the output compare register (OCR) value, thus to provide a 22μs interval the OCR was set to a value of 33.

With a defined interrupt occurrence, it was then required to relate the number of interrupt pulses to the physical angular position of the arm. With respect to the manufacturer pulse specifications presented in Table 6-2, the relationship between interrupt pulse count in software and the known motor shaft positions was plotted as a linear relationship, shown in Figure 6-15.

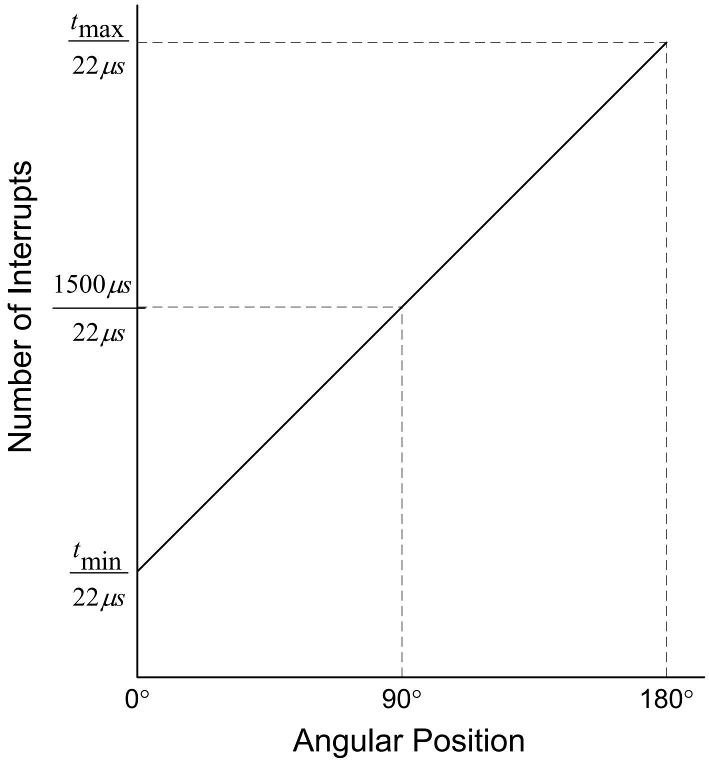


Figure 6-15: Number of Interrupts vs. Angular Position

The six HSR-5995TG servo motors used a pulse of 1100μs to achieve hard left (or 0°) position and 1900μs for hard right (180°). The six HSR-5990TG motors used 600μs hard left and 2400μs for hard right. Both types of motors were positioned at the default reset position of 90° when the pulse was set to 1500μs. This meant that the HSR-5995TG and HSR-5990TG servo motors were positioned at 0° when their interrupt pulse count was set to 50 and 27 interrupts respectively, and were both centred at 90° with a count of 68 interrupts using the configured 22μs interrupt time period. From this linear relationship, it was possible to draw two formulas to determine the number of interrupts required to obtain the required pulse to reach a destination position, provided in degrees.

Table 6-3 presents the pulse data that was determined for the two types of motors from knowledge of a 22μs interrupt at the three defined positions 0°, 90° and 180°. As the default 90° position was considered the most important position, the formulas were based from the interrupt count of 68 to achieve a pulse as close as possible to 1500μs.

HSR-5990TG

θ	0°	90°	180°
# Interrupts	27	68	109
Period(μs)	594	1,496	2,398

HSR-5995TG

θ	0°	90°	180°
# Interrupts	50	68	86
Period(μs)	1,100	1,496	1,892

Table 6-3: Implemented Servo Pulse Timing Data

Using the number of interrupts to generate the given time period defined in Table 6-3, the number of new interrupt counts to create a new pulse for a new desired position provided in degrees was determined using the following formula:

$$Number\_Of\_Interrupt\_Pulses = 68 + K(\theta - 90°)$$

Where the value 68 represents the default 90° position, K is a constant value of the linear gradient, and the acceptable value of the degrees ‘θ’ is a value ranging from 0° through 180°.

The gradient constant K was determined by rearranging the above formula, and using the determined values for each motor between 0° and the 90° position. An example is shown below for the HSR-5990TG motor.

$$27 = 68 + K(0° - 90°)$$

$$K = \frac{27 - 68}{-90} = \frac{41}{90}$$

This process allowed the following formulas to be established.

- HSR-5990TG

$$Number\_Of\_Interrupt\_Pulses = 68 + \frac{41 \times (\theta - 90°)}{90°}$$

- HSR-5995TG

$$Number\_Of\_Interrupt\_Pulses = 68 + \frac{18 \times (\theta - 90°)}{90°}$$

These formulas were then simplified to:

- HSR-5990TG

$$Number\_Of\_Interrupt\_Pulses = \left(\theta \times \frac{41}{90}\right) + 27$$

- HSR-5995TG

$$Number\_Of\_Interrupt\_Pulses = \frac{\theta}{5} + 50$$

After calculating the correct number of pulses required for each servo motor to reach the desired heading, the interrupt was used to either set or clear the output pins. The output pins were simultaneously set high every 20ms as per the defined manufacturer pulse cycle, within the 22μs interrupt. When the pins were set high, the count of the elapsed interrupts was reset to zero to allow the number of elapsed interrupts to be incremented from 0 with each occurrence of the 22μs interrupt thereafter. When the active interrupt count matched the number of required interrupt counts for correct servo positioning using the formula shown above, the individual output pin was cleared, producing a PWM pulse with the required period.

After the successful pulsing technique had been constructed, a method to transition each leg from its current location to the newly desired position was constructed. This was achieved by controlling the rate at which the motor pulse width varied, using ‘current’ and ‘desired’ positions. The rate of change was dictated by modifying a fixed delay in software that occurred between each small transition increment the leg made. This allowed each leg to reach the newly desired position over a series of transitions, over a period of time. This also allowed the frequency at which the incremental/decremental steps were made to be modified to best suit the given motor’s responsibility. For Spike’s operation the base motors needed to rapidly position the leg and joint mechanism, such that the joint motor could then smoothly transition through its larger pulse range, and smoothly drive the leg through its range of motion.

7 RESULTS

7.1 Overview

The aim of this research was to create a novel form of locomotion using a novel robotic design. The robotic design included a navigation system, programmed into its processors that allowed the robot to achieve direction following successfully. By incorporating necessary communication interfaces within the robotic chassis, communication with a desktop PC was achieved to facilitate logging of data as the robot transitioned through its movements. This allowed the internal measurements and calculations of the robot's control systems to be reviewed in real-time by a user, as the robot made decisions based on its environment variables.

The robot was able to relay information using its internal bluetooth wireless interface. This interface acted as a serial pipe for the robot, to allow the real-time monitoring of the navigation system's variables that affected motion. Data such as the position coordinates of the robot and the desired heading were displayed graphically from the transmitted data for analysis and review. The robot's calculations were made from the reading of LDR light sensors to determine its orientation, and the calculated trajectories for each step were graphically presented to ensure correct operation. Wireless communication was required due to the tumbling motion exhibited by the robot. Using a tethered cable would have proven inhibiting to the robot's motion.

The analysis of the robot's ability to move was achieved by comparing transmitted data from the robot with live camera footage of the robot during motion. This data provided the necessary proof of the robot's abilities, and confirmation of its successful operation.

The outcomes from this research are covered in this chapter as follows: first, the chapter will discuss the observations made when the robot took a single step, and will then lead into the robot's ability to take multiple steps at different heading angles. The chapter will next discuss the ability of the robot to determine its orientation based on the light level in its environment, and will finish by considering situations in which the robot's sensors might become inaccurate as a means of determining orientation.

7.2 Single Step

Initially the robot was programmed with a sequence of leg movements that provided the ability for the robot to take a single step. This configuration allowed the robot to move slowly from one orientation to another, such that the transfer of its body mass between legs could be easily seen. The frame by frame images of the robot during the motion of a single step is shown in Figure 7-1.

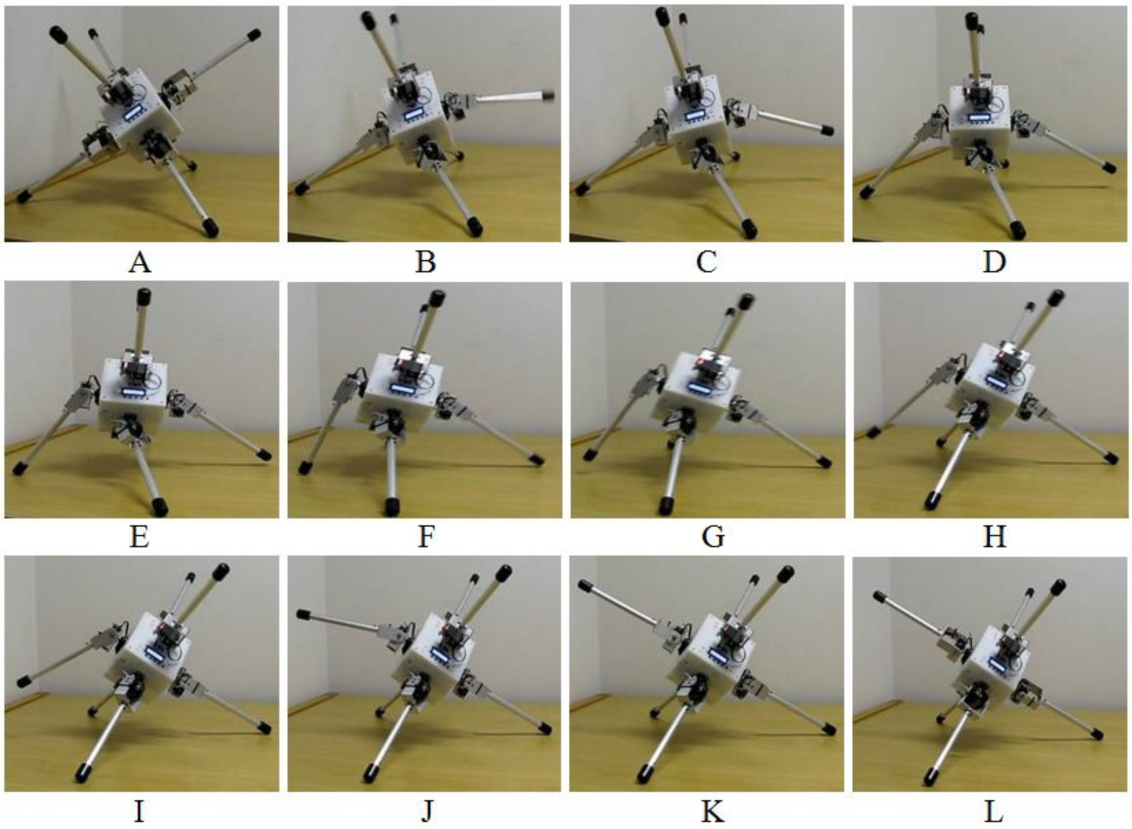


Figure 7-1: Single Step Analysis

In this sequence of leg movements, the robot's ability to travel was created by the use of pulse timing to control the servo motors that dictated the final leg positions. A single step required two 'standing legs' to act as the robot's tipping fulcrum, a 'driving leg' to force the robot's imbalance to occur, and a 'catching leg' that received the weight when the robot fell into the new leg configurations. In frame A of Figure 7-1 the driving leg is shown on the lower left hand side of the frame, the catching leg is shown on the upper right hand side of the frame and the two standing legs remain in contact with the ground at all times throughout all frames of the figure.

For the robot motion to occur, the catching leg must be present on the cube to catch the robot's weight as the robot tips, however the action of lowering the catching leg towards the ground is not required. During the single step example, the catching leg can be seen descending toward the ground surface as the robot moves toward it. This was performed to help reduce the falling impact experienced by the chassis as the robot transitioned from one position to the next.

The first stage of the leg sequence began with determining the orientation of the legs with respect to the ground surface. The LDR light sensors placed within each of the six individual feet were examined, and recordings from each pair of feet on each collinear axis were used to determine which of the legs were up or down. The three legs found to be located on the ground provided information about the robot's orientation, which allowed the determination of the appropriate leg to move to create motion in the required direction. The desired heading trajectory received from the user was compared with the individual leg bearings, and the most appropriate driving leg was selected. The choice for the driving leg was based on the leg's bearing with respect to the desired heading, and whether the leg was placed on the ground surface. After the gathering this information, the first sequence of leg movements for creating the individual step began. This is seen in frames 'A' and 'B' of Figure 7-1, where each leg and joint mechanism was rotated to face the ground surface.

The orientation of the leg and joint mechanism allowed the full torque abilities of each motor to act against the ground surface. In the following stages of the motion, the driving leg's foot was forced toward the ground, in line with the midpoint of the two standing legs and toward the direction of travel. As the driving leg progressed through its range of motion, the centre cube shifted towards a point directly over the two standing legs seen in frame 'F' of the figure. As the driving leg progressed through this range of motion, the catching leg was lowered in preparation to receive the weight of the tumbling robot. In frame 'G', the robot has tipped past its centre of balance and onto the catching leg. After the tumble, all of the legs subsequently returned to a 'reset' position where the servo motors were positioned at 90°, and the legs were once again perpendicular to the cube chassis, seen in the final frame 'L'.

The single step provided proof that the motion concept functioned as planned on the physical robot, and provided an entry point to be expanded into multiple steps toward a direction of choice. The displacement of the robot from its original position was seen from the experiment, which proved the robot's capability of covering a distance using its unique gait. The ability to take subsequent steps required the leg positions and robot orientation to be recalculated after each step, and to be used to calculate the relationship between the robot's position and the desired direction of travel.

7.3 Multiple Steps

To determine the ability of the robot to travel toward a selected heading, several evaluations of the robot took place. Each test required the robot to take several steps by repetition of its single step leg sequence. A testing environment was constructed to allow the robot to move in a controlled situation, and to record the accuracy of the motion. The robot was positioned on the carpeted floor and the heading for the robot to follow was measured and drawn on the floor with a line. The robot was placed appropriately at the beginning of the line to allow a full examination of its motion with respect to following the heading that the line indicated. The robot's motion was filmed by a camera from a fixed, elevated position as it progressed along the desired heading. The data received from the tests was evaluated over multiple steps. In Figure 7-2, a series of camera photos shows the robot position after each of its individual steps along the heading. The 55° heading is indicated by the white line on the carpet which is relative to the robot's starting position facing the camera.

Figure 7-2 demonstrates the robot's ability to take multiple steps to achieve travel in a given direction. As the desired heading does not lie on one of the robot's natural axes of symmetry, the robot was required to make a series of steps that occasionally shifted the robot's centre away from the heading line, in order to maintain travel in the overall required direction of travel. This is shown in frames 'M' & 'N' and 'Q' & 'R'. The robot took steps away from the line of trajectory to maintain positive forward motion along the heading, and made correction steps to adjust for accumulated error, caused by its tripedal stance. The final frame 'X' shows the robot positioned directly over the white indication line at the end of its motion. This demonstrated the ability of the robot to travel in a direction provided by the user over multiple steps. In this example, the robot received information of the desired heading to follow via the wireless remote control. The white line indicated on the ground surface of the testing environment was placed as a reference only.

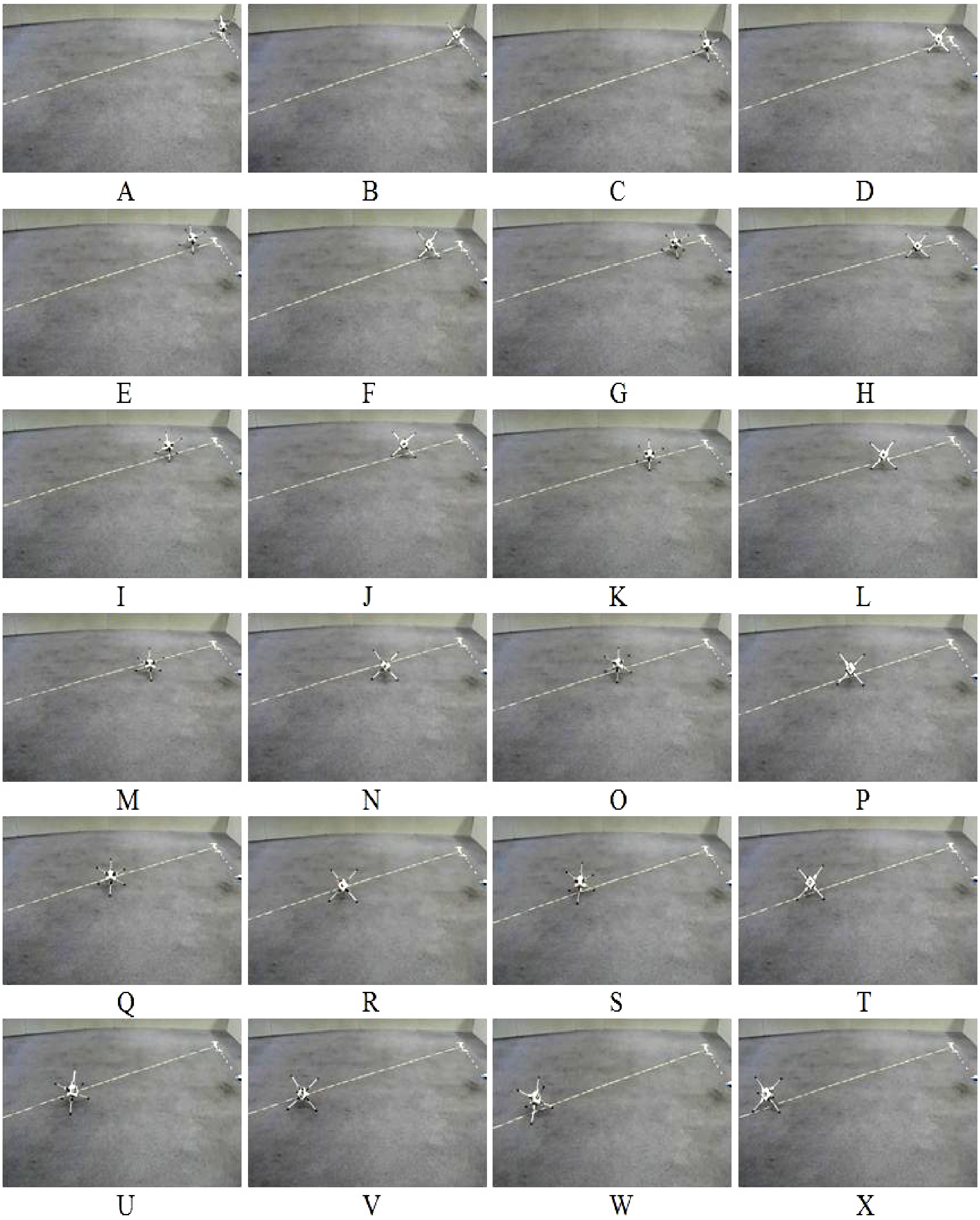


Figure 7-2: Multiple Step Analysis

As the robot tumbled, values from its navigation systems and sensors were transmitted via the robot's wireless bluetooth serial transmitter. These calculation results were captured by a desktop PC and imported into a spreadsheet program. The values provided by the robot allowed a graphical plot of the robot's mathematically calculated positions overlaid with the desired heading.

By using the robot in several tests, an understanding of the navigation system's capacity to handle varying angles of trajectory could be determined. Several tests were conducted, and the results from each were taken and compared with other similar tests. The plotted data included in the graphs of this section are examples of the types of tests and data collected from the robot that provides an indication of the robot's abilities.

The graphs included in the following figures show raw data transmitted from the robot as its motion took place. The bulleted points on the plots are the actual positions of the robot throughout its journey. The linear line is the desired trajectory for the robot to follow.

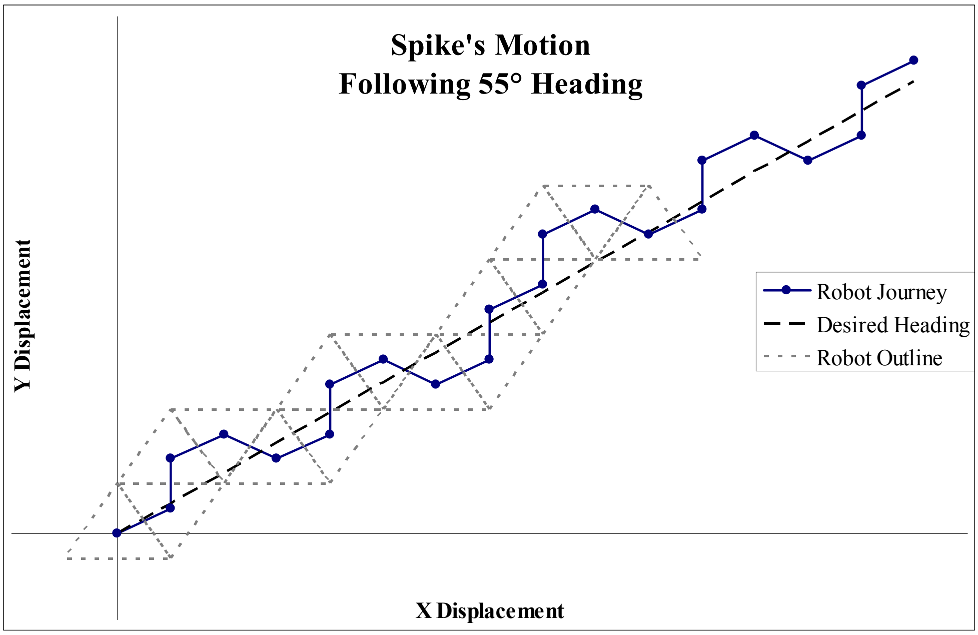


Figure 7-3: Spike's Motion Following 55° Heading

The graph shown in Figure 7-3 shows the data from the robot travelling along a 55° heading. The position of the robot in frames 'M' & 'N' of its motion in Figure 7-2, where the robot was physically located away from the trajectory line, is clearly visible in the plot shown in Figure 7-3. The 55° heading was chosen as it is close 60°, a bearing that lies on one of the robots natural axes of symmetry. Using a heading slightly less than one of the robot's natural axis of symmetry allowed the robot to clearly demonstrate its capacity to adjust for the accumulated errors, caused by the robot's tripedal stance. This is seen as an increased number of steps toward a northern direction, before adjusting east as a corrective manoeuvre.

The robot was placed in additional tests where the desired heading was adjusted. The measurements taken from several tests were graphed for analysis. The data collected from the robot corresponded with the robot's physical performance during each of the tests. A mix of trajectories was used to demonstrate the ability of the robot to follow headings that lay on and off the robot's natural axes.

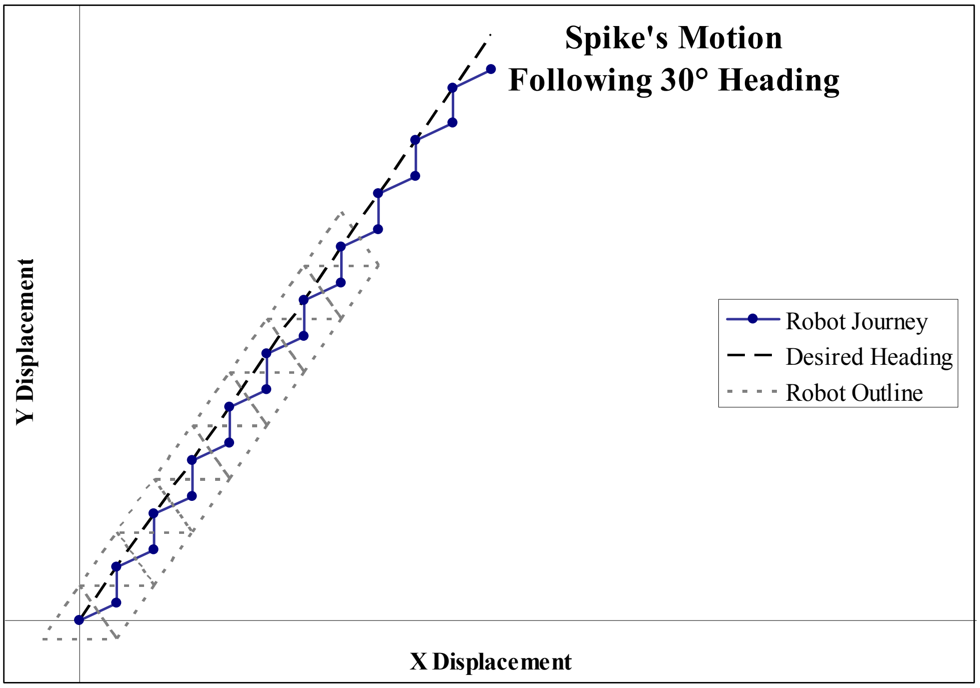


Figure 7-4: Spike's Motion Following 30° Heading

In Figure 7-4, a graph showing 21 steps of Spike's motion following a heading of 30° is shown. This heading lay on one of the robot's natural axes of symmetry, and hence the robot followed a periodic motion that did not require corrective steps to be taken. Over the entire motion, the desired heading line lay within the triangular region defined by the three legs located on the ground surface. The plot matches the motion physically exhibited by the robot, as the transmitted coordinate data seen in the figure is transmitted in real time as the robot navigates along the heading.

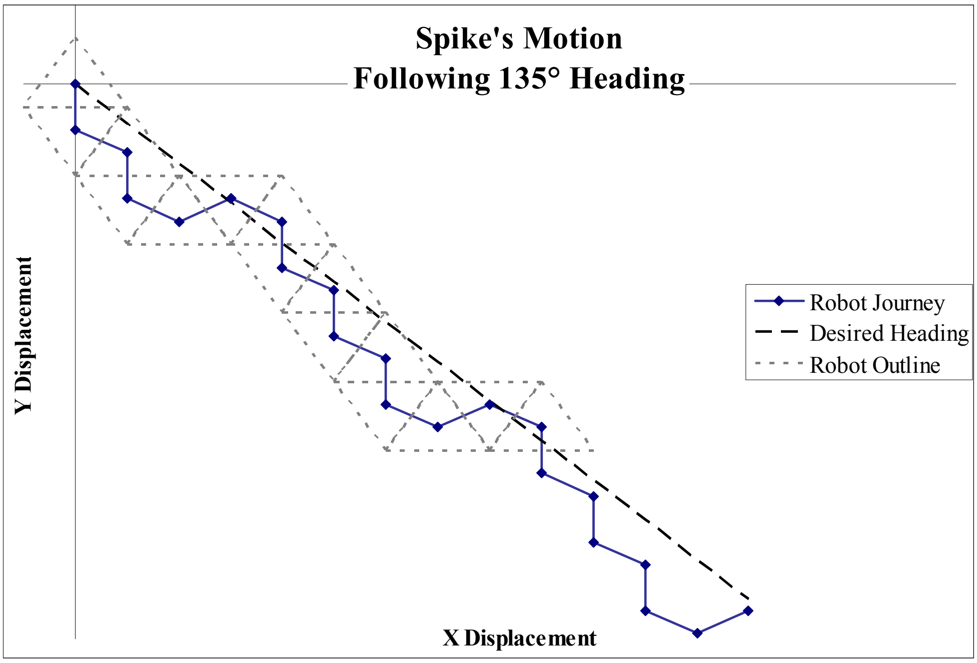


Figure 7-5: Spike's Motion Following 135° Heading

The graph shown in Figure 7-5 provides graphical representation of the robot's journey during travel along a 135° heading. The plotted data shows that the robot took several steps south of the required direction before making a correction step in an easterly direction, which positioned the robot closer to the desired heading. By attempting travel with a 135° heading, the flexibility of the robot's navigation system to make decisions of which single leg to drive could be reviewed. The 135° heading lay away from any of the natural axes of the robot and hence, a selection of steps was required to achieve travel along this heading. As the imaginary target was moved further away from the robot's actual position, the navigation system performed an increased number of southward steps allowing the robot to maintain the overall desired heading. When a smaller robot-to-target distance was specified, more frequent movements about the heading line could be seen. The dynamic adjustment by the robot was seen in physical testing of the robot, and the figure demonstrates the robot's ability to maintain overall travel in the desired direction over multiple steps.

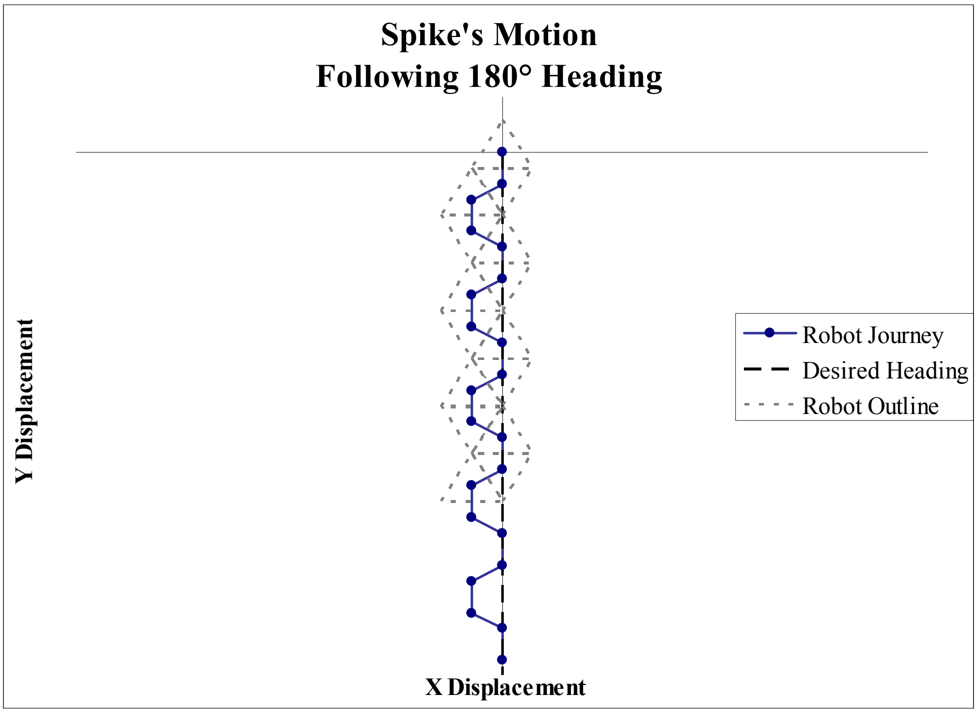


Figure 7-6: Spike's Motion Following 180° Heading

The graph shown in Figure 7-6 shows the robot's journey during travel along a 180° heading. This heading lies directly behind the robot in its starting position, and hence lies on one of the six axes of symmetry for the robot. The robot is able to adjust its steps between two directions, southerly and south westerly without the need for several steps in a single direction that the robot must correct over time. The western direction is chosen by the navigation system due to the calculations and definitions of regions made by the navigation system. The robot's motion along this heading allowed a high degree of accuracy to be achieved for the robot's journey, during which the robot was never more than half its width from the trajectory line.

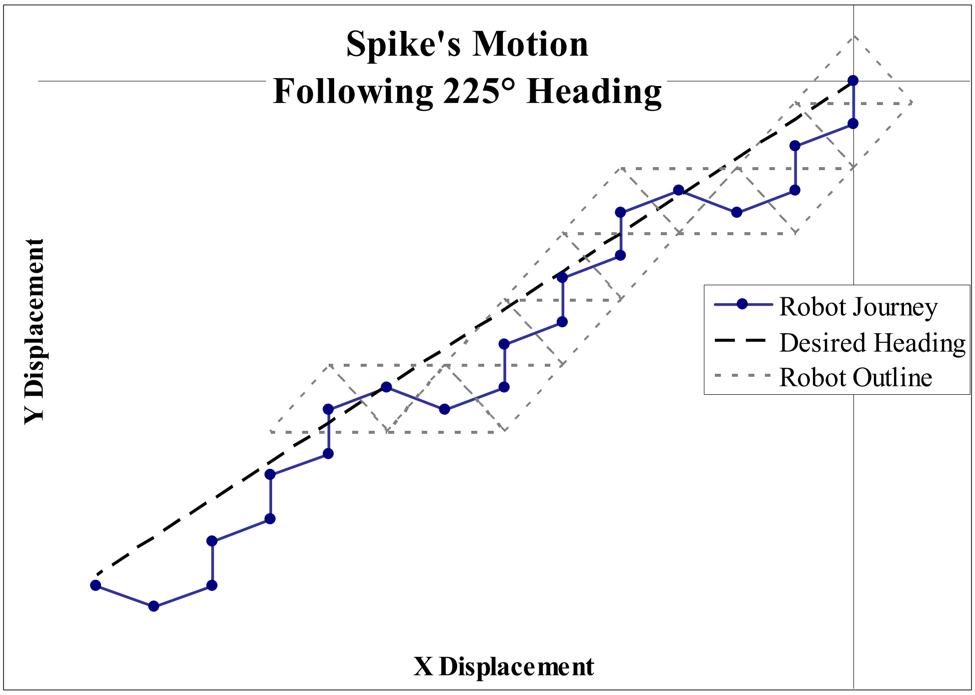


Figure 7-7: Spike's Motion Following 225° Heading

The graph shown in Figure 7-7 shows the robot's journey along on a 225° heading. The motion of the robot resembles the robot's motion in the plot of travel in 135° in Figure 7-5. Figure 7-7 indicates that the robot took several steps south of the trajectory heading, with westerly correction steps made to adjust the robot's position. The robot was able to achieve motion along the given trajectory, with the data shown in the figure providing verification of the robot's physical motion during the test.

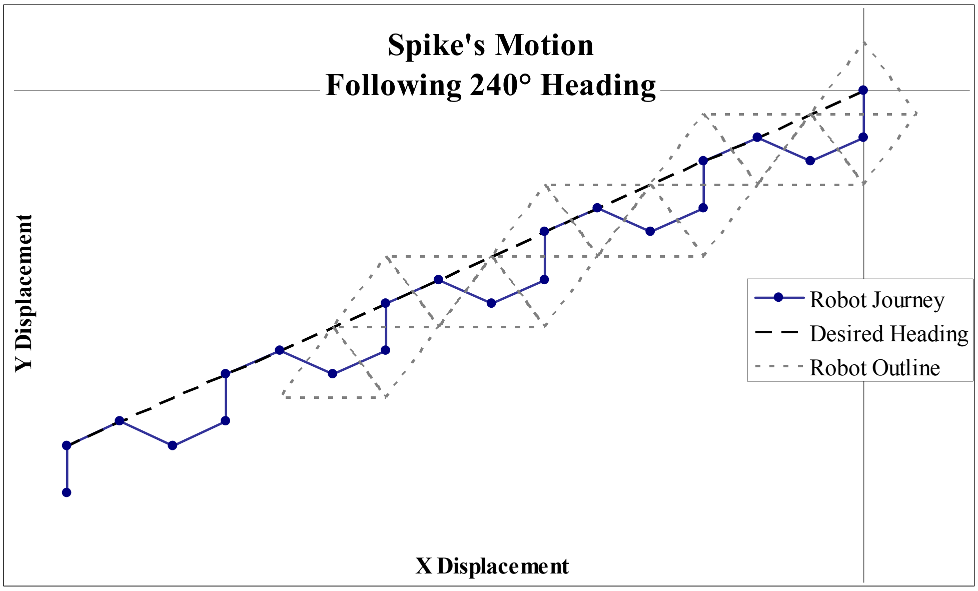


Figure 7-8: Spike's Motion Following 240° Heading

The graph shown in Figure 7-8 shows the robot's journey along one of its three axes of symmetry. The robot was able to follow the 240° heading using a sequence of leg movements that allowed the robot to position itself no greater than half its width from the trajectory line. This test provided proof that the robot was able to travel at a heading that lay on one of the robot's axes. Observations of the robot's journey are similar to the motion exhibited in Figure 7-6, in which the robot can also be seen to follow a heading that lay on an axis of symmetry of the robot.

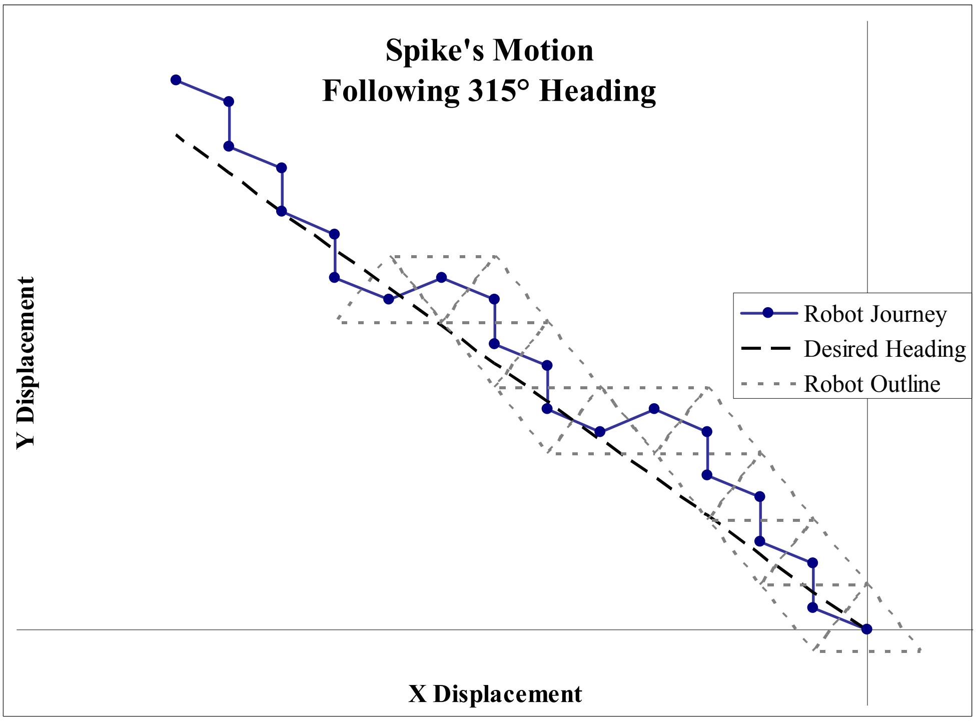


Figure 7-9: Spike's Motion Following 315° Heading

The graph shown in Figure 7-9 confirmed the operation of the robot following a trajectory of 315°, which lay in a north westerly direction from the robot's starting position. The robot took several steps north of the required direction before making a westerly correction step to adjust its position with respect to the target trajectory. The plot of the robot's motion provided graphical confirmation of what was visually exhibited by the robot during the test.

The robot's motion shown in these figures demonstrates that the ability of the robot to follow a given heading improves as the robot made an increased number of steps in the given direction. Small variations were always present in the robot's motion, however, given the fixed triangular geometry of the robot's footprint, the robot was able to adjust for its physical limitations with a high degree of accuracy.

As the operation of the navigation system was to maintain positive motion toward a 'target' position, the robot was required to occasionally transition away from the trajectory heading over several steps, before taking corrective action. This feature of the system ensured that the motion of the robot maintained forward direction along the trajectory heading, while allowing the robot's position to maintain close proximity to the desired heading with corrective steps. This was facilitated by maintaining a fixed "robot-to-target" distance constant within the setup of the navigation system. An increase in this distance made the robot take an increased number of steps away from the trajectory heading, which meant the heading was still followed but over a much greater number of steps. Using a shorter distance constant, the robot had more frequent corrective steps to the trajectory heading whilst maintaining forward motion and was therefore the preferred option.

The transmitted data from the robot matched the physical operation of the robot in all testing situations, providing an appropriate numerical confirmation of the robot's abilities. The data presented in the figures showed raw data from the robot as it transitioned through each sequence of movements. The outline of the robot's footprint was added as an indicative assistance to show the reason for the direction choice the robot made for each step.

7.4 Identification of the Ground Surface

To determine the ability of the robot to make decisions about its orientation in its environment, the robot was placed in many different lighting conditions and the light measurements were recorded. These measurements provided an understanding of the robot's ability to correctly determine which of the legs were facing upward or downward. The robot was placed in both outdoor and indoor settings where the light readings measured by the robot were recorded and assessed. These tests allowed an understanding to be developed of the limitations the robot would experience when attempting to move in light conditions that differ from the testing environment. Twelve separate tests were performed and the readings were noted. The light measurements recorded were measured as an 'up' reading, where the intensity of light shining directly downward was measured, and a 'down' measurement that indicated the intensity of reflected light when the sensor was placed at 45° to the ground surface. Measurements performed in this way allowed the measurement of the leg sensor's responses when facing upward and downward after each step respectively.

The measurements recorded from the light sensors within the legs represented interpreted values of the light experienced by each of the legs. This light was converted by the internal electronics of the robot to a numerical value. An extreme bright light condition represented a full scale reading of 255 and an extreme low light condition was displayed as 0. The recorded measurements showed variations between different environments that were affected by the ground surface colour and its light reflective ability. The colour of the ground surface was different in each test and this also affected the amount of light reflected from the ground into the leg sensors. By taking several measurements of the different colours of carpet in an indoor environment, the variation of reflected light from carpet colour could be determined. The results of the tests are tabulated in this section. Note that legs 1 and 6 share and axis, legs 2 and 4 share and axis and legs 3 and 5 also share and axis.

Surface Type	Carpet - Light Grey				
Measured Light Level (lux)					
UP	540				
DOWN	40				
Leg Readings in Starting Orientation					
Leg 1	*Leg 2*	*Leg 3*	*Leg 4*	*Leg 5*	*Leg 6*
240	175	180	230	245	180

Table 7-1: Light Measurements on Light Grey Carpet

Surface Type	Carpet - Dark Blue				
Measured Light Level (lux)					
UP	450				
DOWN	25				
Leg Readings in Starting Orientation					
Leg 1	*Leg 2*	*Leg 3*	*Leg 4*	*Leg 5*	*Leg 6*
235	110	170	248	240	139

Table 7-2: Light Measurements on Dark Blue Carpet

Surface Type	Carpet - Light Blue				
Measured Light Level (lux)					
UP	800				
DOWN	80				
Leg Readings in Starting Orientation					
Leg 1	*Leg 2*	*Leg 3*	*Leg 4*	*Leg 5*	*Leg 6*
240	200	190	240	220	190

Table 7-3: Light Measurements on Light Blue Carpet

Surface Type	Carpet - Dark Green				
Measured Light Level (lux)					
UP	280				
DOWN	18				
Leg Readings in Starting Orientation					
Leg 1	*Leg 2*	*Leg 3*	*Leg 4*	*Leg 5*	*Leg 6*
190	100	154	240	230	125

Table 7-4: Light Measurements on Dark Green Carpet

Surface Type	Carpet - Grey - Low Ambient Light				
Measured Light Level (lux)					
UP	20				
DOWN	4				
Leg Readings in Starting Orientation					
Leg 1	*Leg 2*	*Leg 3*	*Leg 4*	*Leg 5*	*Leg 6*
115	20	32	140	145	50

Table 7-5: Light Measurements on Grey Carpet with Low Ambient Light

Surface Type	White Lino - Shaded Area				
Measured Light Level (lux)					
UP	85				
DOWN	40				
Leg Readings in Starting Orientation					
Leg 1	*Leg 2*	*Leg 3*	*Leg 4*	*Leg 5*	*Leg 6*
180	160	180	175	195	165

Table 7-6: Light Measurements on White Lino - Shaded Area

Surface Type	White Lino - Well Lit Area				
Measured Light Level (lux)					
UP	340				
DOWN	110				
Leg Readings in Starting Orientation					
Leg 1	*Leg 2*	*Leg 3*	*Leg 4*	*Leg 5*	*Leg 6*
225	220	225	190	240	210

Table 7-7: Light Measurements on White Lino in a Bright Area

Surface Type	Tiles - Grey & Slightly Reflective				
Measured Light Level (lux)					
UP	2100				
DOWN	850				
Leg Readings in Starting Orientation					
Leg 1	*Leg 2*	*Leg 3*	*Leg 4*	*Leg 5*	*Leg 6*
240	211	222	220	253	210

Table 7-8: Light Measurements on Grey Tiles

Surface Type	Tarmac Pavement Outside @5.30pm Sunny Day				
Measured Light Level (lux)					
UP	48000				
DOWN	23000				
Leg Readings in Starting Orientation					
Leg 1	*Leg 2*	*Leg 3*	*Leg 4*	*Leg 5*	*Leg 6*
255	254	253	255	255	251

Table 7-9: Light Measurements on Tarmac Pavement on Sunny Day

Surface Type	Tarmac Pavement Outside @5.30pm Cloudy Day				
Measured Light Level (lux)					
UP	7500				
DOWN	900				
Leg Readings in Starting Orientation					
Leg 1	*Leg 2*	*Leg 3*	*Leg 4*	*Leg 5*	*Leg 6*
250	240	246	252	254	241

Table 7-10: Light Measurements on Tarmac Pavement on Cloudy Day

Surface Type	Concrete - Cloudy Day 6pm				
Measured Light Level (lux)					
UP	15000				
DOWN	2000				
Leg Readings in Starting Orientation					
Leg 1	*Leg 2*	*Leg 3*	*Leg 4*	*Leg 5*	*Leg 6*
255	250	251	254	255	249

Table 7-11: Light Measurements on Concrete on Cloudy Day

Surface Type	Grass - Cloudy Day 6pm				
Measured Light Level (lux)					
UP	19000				
DOWN	350				
Leg Readings in Starting Orientation					
Leg 1	*Leg 2*	*Leg 3*	*Leg 4*	*Leg 5*	*Leg 6*
255	45	65	252	255	45

Table 7-12: Light Measurements on Grass on Cloudy Day

In these experiments legs 2, 3 and 6 were placed on the ground surface. This orientation was defined in the robot's navigation system as the starting orientation. These legs should have had a low reading of light when compared with the other legs, to clearly indicate how the robot was orientated with the ground surface. The two feet on each of the robot's three separate axes of symmetry were compared with each other to determine which of the three feet were placed on the ground.

Determining the robot’s orientation by comparison of the light levels found on a single axis for each of the robot’s three axes, was found to be the most effective method for determining the robot’s orientation. The results from these tests showed how the sensors were able to detect light in different lighting conditions and surface types such that validity of the robot’s decisions about its orientation in these environments was established.

The outcome from the light detection tests showed that the light sensors were best suited to indoor environments with even lighting, where the floor was carpeted. Although favourable results were obtained from an outdoor grass type surface, the results for this particular test may be inaccurate. It is possible that the individual blades of grass work to obscure the grounded light sensors. Therefore in a situation where the grass height differed, the light measurements for the robot may be inconsistent. In outdoor environments, it may be desirable to use a different type of ground detection sensor, such as a pressure sensor or accelerometer.

From analysis of the test data, floor reflection and floor colour influenced the level of light detected by the light sensors. This was due to the amount of light reflected from the ground surface onto the LDR light sensors. Using a lino or tile surface, the light was more easily reflected back onto the light sensors on the grounded legs, and the robot was more likely to make wrong estimations about its orientation. The experiments carried out on the non-reflective surfaces, such as carpet, provided the greatest differences between light and shade, and these were consequently found to be the easiest surfaces for the robot to determine its orientation.

8 CONCLUSIONS

The primary goal of the research in this thesis was to develop a novel form of robotic motion on a unique robotic platform, and this has been achieved. As this aim included two primary sections, robotic development and creation of motion, the two primary areas should be examined separately.

The robotic development began with a basic model, and was subsequently developed into a successfully designed and constructed robot, providing a robust test platform for development and research. The robotic chassis was successfully fabricated using a plastics 3D rapid prototyping printer, and the leg assemblies and joint mechanisms operated as intended, as discussed in the design methodology. The custom designed electronics allowed the robot to pass information between its dual processors, and the system was equipped with batteries allowing the robot to operate wirelessly during testing. The hardware was verified as working, and the software built on the electrical systems was also verified as working. The modular structure of the electrical design was developed with the intention of growth and expansion into new areas of processing, and sensor development, which may also extend into alternative methods of motion. The electrical systems were fully integrated into the mechanical design, utilising the available free space contained within the chassis design, maximising the robot's abilities and achieving a finished and professionally designed product.

The novel form of motion was successfully created using the unique robot design. The robot operated using a series of tumbling and falling actions to create movement along a specified heading. The robots performance was observed in a number of test situations that showed how the robot was able to manipulate its six legs effectively. The inherent stability posed by the robot's tripedal stance was overcome using a method of shifting the robot's centre of mass toward a tipping point, such that the robot fell into new leg configurations and robot orientations. The ability for the legs to be driven to cause the robot to tumble was achieved using servo motors for actuation, and using various sensors built into the robot design. The LDR light sensors were successfully able to inform the robot of its current orientation with respect to the ground, so that dynamic recalculation of the trajectory heading could be achieved for navigation. The capacity of the robot to determine its orientation in a range of light levels was analysed, which led to the conclusion that the robot's optimal environment for operation is in moderate and even lighting with a non-reflective carpet type surface. Directing the robot to travel toward a destination heading instigated by the remote control, represents the first generation of the robot’s control system. It is intended that further development of existing systems will allow more advanced control methods to be explored in the future.

The robot’s navigation system operated using a mathematical algorithm developed in software, which used a bearing, provided from the remote control, to direct the robot along a heading by maintaining a fixed distance between the robot and an imaginary target on the heading trajectory. The algorithm that was used in the robot’s navigation system formed the stepping strategy for the robotic motion. The optimum direction for the robot to tip toward was found using an analysis of knowledge of the robot’s geometry, and the desired heading for the robot movement. The triangular footprint of the robot created a challenge for the navigation system to overcome, but was accomplished using several fluctuation steps about a heading trajectory line as the robot moved. It was found that reducing the distance between the robot and the imaginary target caused a higher frequency of fluctuation about the trajectory heading, but allowed the robot to maintain a closer average distance to the trajectory heading, causing more precise motion. The overall accuracy of the robot to follow the given heading was found to be reliant on the number of steps taken by the robot, while over time, the robot was able to self correct for accumulated error with a sideways step. An increased number of steps taken by the robot allowed the overall accuracy of motion to improve with respect to the robot’s starting position.

The emphasis in this thesis has been placed on describing the research, design and construction of the unique robotic hardware, and the process of creating navigation for the robot to achieve its goal of novel and unique motion. The robot’s symmetry and triangular footprint were discussed with multiple scenarios that demonstrate the operation of the robot. Experiments carried out on the robot demonstrate the performance of the motion in a controlled environment, and a frame-by-frame example of the motion is included.

In conclusion, the research findings of this thesis were: a novel form of robotic motion, using a custom designed and built robotic design. The development in this research achieved the desired form of motion and successfully reached the thesis objectives.

9 FUTURE DEVELOPMENT

The robotic development in this research consisted of many aspects that led to a successful and operational prototype. The gait developed on the unique hardware was an example of a novel form of robotic motion, and it is intended that the hardware will offer a platform for new methods of control and motion to be developed and evaluated.

The tripedal stance of Spike poses numerous challenges for the design of new gaits. Alternative control systems for Spike may be developed with the aid of a fully featured computer simulation model of the robot. Using a computer model may allow numerous tests to be performed rapidly, with the intention of evaluating final simulations in physical reality. This may allow creation and modelling of behaviours for different robot environments to be easily established.

In future work, a new type of gait may be developed on Spike, which could be created by using a genetic algorithm [90] or other evolutionary technique. Genetic algorithms applied to a simulation of Spike on a PC may allow the creation of a gait to provide motion using a method that may not be easily conceivable by humans. The effectiveness of the genetic algorithm's optimal solution may be evaluated in physical reality and compared with the manually programmed gait developed in this thesis. The robot created in this development is intended to provide the necessary electrical and mechanical interfaces to allow future development of the robot's abilities.

Other suitable control methods for manipulating the robot's leg positions may be developed. As each servo motor may be defined to reach an angular position, future development may also include the design of a kinematics system to determine final leg position, for each of the robot's six legs. The hardware included in Spike is intended to provide the necessary processing and communication abilities, to allow successful evaluation of gaits developed using both the robot's processing abilities as well as the processing abilities of a desktop PC.

Developments of new types of leg sensors may also expand upon the range of areas where the robot may be used. The LDR sensors have the connection ability to be changed for pressure or sound sensors for evaluation. The modularity of the robotic design facilitates this development.

Including a three axis accelerometer into the robotic design may allow an alternative method of determining robot orientation to be realised. The benefit of using accelerometer technology may allow limitations posed by light or pressure sensors to be overcome. The accelerometer may supply sufficient information about the robot's orientation without the need for an additional external factor such as light.

It is also intended that the electrical interface included in the robotic design for changing the control daughterboard, allows for a range of data processing facilities that may extend beyond the current processing abilities of the robot. By changing processing technology, the control system may be enhanced allowing the development of more advanced robot capability. Such processing technologies may include an ARM processor or FPGA technology.

Another area for future development is the adaptation of the existing motion control system to facilitate travel more closely aligned to the specified heading. The triangular footprint used as the basis of the navigation system of the robot, could be modified by dynamic adjustment of the robot's leg positions on the ground. By adjusting the leg positions on the ground, irregular shaped triangular footprints may be created that may allow an increased number of directions for the robot to travel toward, rather than the current three directions separated evenly by 120°. Programming this idea would require significant software construction around the dynamic reconfiguration of the leg positions in real-time, and would present a new challenge of robot stability versus robot direction selection.

APPENDICIES

Appendix A *Microcontroller Pin Configuration*

Spike the Robot

Pin Description Information

Daughter Board Connections - Mega 128

	Connected To	Description	I/O µC	Power Source	Connector
PA.0	LCD D0	Data lines to LCD screen	Output	3.3V/5V	2 B18
PA.1	LCD D1	Data lines to LCD screen	Output	3.3V/5V	2 B19
PA.2	LCD D2	Data lines to LCD screen	Output	3.3V/5V	2 B20
PA.3	LCD D3	Data lines to LCD screen	Output	3.3V/5V	2 B21
PA.4	LCD D4	Data lines to LCD screen	Output	3.3V/5V	2 B22
PA.5	LCD D5	Data lines to LCD screen	Output	3.3V/5V	2 B23
PA.6	LCD D6	Data lines to LCD screen	Output	3.3V/5V	2 B24
PA.7	LCD D7	Data lines to LCD screen	Output	3.3V/5V	2 B25
PB.0	SPI SS	SPI Communication	Bi-Direction	3.3V	1 B9
PB.1	SPI SCK	SPI Communication	Bi-Direction	3.3V	1 B10
PB.2	SPI MOSI	SPI Communication	Bi-Direction	3.3V	1 B11
PB.3	SPI MISO	SPI Communication	Bi-Direction	3.3V	1 B12
PB.4	Spare P20.5	Connected to Header pins on Motherboard P20	Bi-Direction	3.3V	P20
PB.5	Spare P20.4	Connected to Header pins on Motherboard P21	Bi-Direction	3.3V	P20
PB.6	Spare P20.3	Connected to Header pins on Motherboard P22	Bi-Direction	3.3V	P20
PB.7	Spare P20.2	Connected to Header pins on Motherboard P23	Bi-Direction	3.3V	P20
PC.7	User Button 3	User Button Active High	Input	3.3V	2 A17
PC.6	User Button 4	User Button Active High	Input	3.3V	2 A18
PC.5	Remote Control Power Switch for SPI method	Turns on Voltage translators that allow SPI communication	Output	3.3V	2 A19
PC.4	LCD Backlight Power Switch	Allows the LCD backlight to operate. Connected to Low Side MOSFET	Output	5V	2 A20
PC.3	LCD Power Switch	Power for LCD Screen. Connected to Low Side MOSFET	Output	5V	2 A21
PC.2	LCD RS	LCD Screen Register Select	Output	3.3V/5V	2 A22
PC.1	LCD R/W	LCD Screen Read/Write	Output	3.3V/5V	2 A23
PC.0	LCD E	LCD Enable (ECLK)	Output	3.3V/5V	2 A24
PD.0	Bluetooth Power Switch	Power for Bluetooth module. Connected to Low Side MOSFET	Output	3.3V	1 B24
PD.1	Zigbee Power Switch	Power for Zigbee module. Connected to Low Side MOSFET	Output	3.3V	1 B23
PD.2	Zigbee Tx	Zigbee data transmit line to Microcontroller (µC Rx1)	Input	3.3V	1 B21
PD.3	Zigbee Rx	Zigbeeh data Recieve line to Microcontroller (µC Tx1)	Output	3.3V	1 B22
PD.4	Remote Control Ack	Remote Control Acknowledge line	Input	3.3V/5V	1 B8
PD.5	Remote Control CLK	Remote Control Clock line	Output	3.3V/5V	1 B7
PD.6	Remote Control ATI	Remote Control Attention line	Output	3.3V/5V	1 B6
PD.7	Remote Control Cmd	Remote Control Command line	Output	3.3V/5V	1 B5
PE.0	Bluetooth Tx	Bluetooth data transmit line to Microcontroller (µC Rx0)	Input	3.3V	1 B1
PE.1	Bluetooth Rx	Bluetooth data receive line from Microcontroller (µC Tx0)	Output	3.3V	1 B2
PE.2	Remote Control Power Switch for BitBang method	Turns on Voltage translators that allow bitbang communication	Output	3.3V	1 B3
PE.3	Remote Control Data	Remote Control Data line	Input	3.3V/5V	1 B4
PE.4	Spare P20.7	Connected to Header pins on Motherboard P21	Bi-Direction	3.3V	P20
PE.5	Spare P20.6	Connected to Header pins on Motherboard P22	Bi-Direction	3.3V	P20
PE.6	Spare P20.1	Connected to Header pins on Motherboard P23	Bi-Direction	3.3V	P20
PE.7	Remote Control Power Switch	Connected to Low side MOSFET of Remote Control Gnd	Output	3.3V/5V	1 B13
PF.0	ADC0 P19.1	Connected to Header pins on Motherboard P19	Bi-Direction	3.3V	2 B1
PF.1	ADC1 P19.2	Connected to Header pins on Motherboard P19	Bi-Direction	3.3V	2 B2
PF.2	ADC2 P19.3	Connected to Header pins on Motherboard P19	Bi-Direction	3.3V	2 B3
PF.3	ADC3 P19.4	Connected to Header pins on Motherboard P19	Bi-Direction	3.3V	2 B4
PF.4	JTAG TCK	JTAG Programming	Bi-Direction	3.3V	J1 & 1 A20
PF.5	JTAG TMS	JTAG Programming	Bi-Direction	3.3V	J1 & 1 A21
PF.6	JTAG TDO	JTAG Programming	Bi-Direction	3.3V	J1 & 1 A19
PF.7	JTAG TDI	JTAG Programming	Bi-Direction	3.3V	J1 & 1 A22
PG.0	Spare P21.2	Connecte to Header pins on Motherboard P21	Bi-Direction	3.3V	2 A25
PG.1	Spare P21.3	Connecte to Header pins on Motherboard P21	Bi-Direction	3.3V	2 A26
PG.2	Spare P21.1	Connecte to Header pins on Motherboard P21	Bi-Direction	3.3V	2 B26
PG.3	User Button 1	User Button Active High	Input	3.3V	1 B26
PG.4	User Button 4	User Button Active High	Input	3.3V	1 B25

Spike the Robot

Pin Description Information

Mother Board

	Connected To	Description	I/O µC	Power Source
PA.0	Servo 1	Position Pulse for Servo	Output Pulse	Source 1
PA.1	Servo 2	Position Pulse for Servo	Output Pulse	Source 1
PA.2	Servo 3	Position Pulse for Servo	Output Pulse	Source 1
PA.3	Servo 4	Position Pulse for Servo	Output Pulse	Source 1
PA.4	Servo 5	Position Pulse for Servo	Output Pulse	Source 2
PA.5	Servo 6	Position Pulse for Servo	Output Pulse	Source 2
PA.6	Servo 7	Position Pulse for Servo	Output Pulse	Source 2
PA.7	Servo 8	Position Pulse for Servo	Output Pulse	Source 2
PB.0	SPI SS	SPI Communications to Daughter Board	Bi-Direction	3.3V Always ON
PB.1	SPI SCK	SPI Communications to Daughter Board	Bi-Direction	3.3V Always ON
PB.2	SPI MOSI	SPI Communications to Daughter Board	Bi-Direction	3.3V Always ON
PB.3	SPI MISO	SPI Communications to Daughter Board	Bi-Direction	3.3V Always ON
PB.4	Multiplexer A0	Analouge Multiplexer Address Bit	Output	3.3V Always ON
PB.5	Multiplexer A1	Analouge Multiplexer Address Bit	Output	3.3V Always ON
PB.6	Multiplexer A2	Analouge Multiplexer Address Bit	Output	3.3V Always ON
PB.7	Multiplexer A3	Analouge Multiplexer Address Bit	Output	3.3V Always ON
PC.0	Servos Power Switch	Turns on the High Current MOSFETs (3) for Power Supply to All Servos	Output	3.3V & 5V
PC.1	Battery 1 Diagnostics	From High Current MOSFET 1. Allows fault conditions to be detected	Input	3.3V
PC.2	Battery 2 Diagnostics	From High Current MOSFET 2. Allows fault conditions to be detected	Input	3.3V
PC.3	Battery 3 Diagnostics	From High Current MOSFET 3. Allows fault conditions to be detected	Input	3.3V
PC.4	Servo 9	Position Pulse for Servo	Output Pulse	Source 3
PC.5	Servo 10	Position Pulse for Servo	Output Pulse	Source 3
PC.6	Servo 11	Position Pulse for Servo	Output Pulse	Source 3
PC.7	Servo 12	Position Pulse for Servo	Output Pulse	Source 3
PD.0	On/Off Button	User On/Off Button Active Low	Input	3.3V Always ON
PD.1	Motherboard Power Switch	Connected to Enable pin of Voltage Regulators on Motherboard.	Output	3.3V Always ON
PD.2	Buzzer Switch	Connected to Mosfet to conrtol Buzzer	Output	5V
PD.3	LED2 Green	Controls LED	Output	3.3V Always ON
PD.4	LED3 Green	Controls LED	Output	3.3V Always ON
PD.5	LED4 Green	Controls LED	Output	3.3V Always ON
PD.6	LED1 Red	Controls LED	Output	3.3V Always ON
PD.7	LED1 Green	Controls LED	Output	3.3V Always ON
PE.0	Spare P3.1	Connected to Header pins on Mother Board P3	Bi-Direction	3.3V Always ON
PE.1	Spare P3.2	Connected to Header pins on Mother Board P3	Bi-Direction	3.3V Always ON
PE.2	Spare P3.3	Connected to Header pins on Mother Board P3	Bi-Direction	3.3V Always ON
PE.3	Spare P3.4	Connected to Header pins on Mother Board P3	Bi-Direction	3.3V Always ON
PE.4	Spare P3.5	Connected to Header pins on Mother Board P3	Bi-Direction	3.3V Always ON
PE.5	LED2 Red	Controls LED	Output	3.3V Always ON
PE.6	LED3 Red	Controls LED	Output	3.3V Always ON
PE.7	LED4 Red	Controls LED	Output	3.3V Always ON
PF.0	Multiplexer ADC.0	Output from Multiplexer. 6 values of LDRs, 4 values of Batteries	Input	3.3V
PF.1	Temp ADC.1	Output from On board temperature sensor	Input	3.3V
PF.2	Not Connected		N/A	
PF.3	Not Connected		N/A	
PF.4	JTAG TCK	JTAG Programming	Bi-Direction	3.3V Always ON
PF.5	JTAG TMS	JTAG Programming	Bi-Direction	3.3V Always ON
PF.6	JTAG TDO	JTAG Programming	Bi-Direction	3.3V Always ON
PF.7	JTAG TDI	JTAG Programming	Bi-Direction	3.3V Always ON
PG.0	Not Connected		N/A	
PG.1	Not Connected		N/A	
PG.2	Not Connected		N/A	
PG.3	Not Connected		N/A	
PG.4	Not Connected		N/A	

REFERENCES

[1] C. Coyte, M. Beckerleg, and J. Collins, "Spike: A Six Legged Cube Style Robot," in *Proceedings of the 2nd International Conference on Intelligent Robotics and Applications* Singapore: Springer-Verlag, 2009, pp. 535-544.

[2] Y. N. Shimon, *Handbook of Industrial Robotics*, 2nd Edition ed.: John Wiley & Sons, Inc, 1999.

[3] E. Garcia, M. A. Jimenez, P. G. De Santos, and M. Armada, "The Evolution of Robotics Research," *Robotics & Automation Magazine, IEEE,* vol. 14, pp. 90-103, March 2007.

[4] L. Eung Chang, C. Hyun Do, K. Soo Hyun, and K. Yoon Keun, "Floor-types Identification Method for Wheel Robot using Impedance Variation," in *International Conference on Control, Automation and Systems*, Seoul, Korea, 2008, pp. 2340-2343.

[5] J. L. Jones, "Robots at the Tipping Point: The Road to iRobot Roomba," *IEEE Robotics & Automation Magazine,* vol. 13, pp. 76-78, March 2006.

[6] F. Carreira, T. Canas, A. Silva, and C. Cardeira, "i-Merc: A Mobile Robot to Deliver Meals inside Health Services," in *IEEE Conference on Robotics, Automation and Mechatronics*, Bangkok, 2006, pp. 1-8.

[7] J. Gaston, K. Raahemifar, and P. Hiscocks, "A Cooperative Network of Reconfigurable Stair-Climbing Robots," in *IEEE International Symposium on Circuits and Systems*, Island of Kos, 2006, p. 4 pp.

[8] L. E. Parker, "ALLIANCE: An Architecture for Fault Tolerant, Cooperative Control of Heterogeneous Mobile Robots," in *Proceedings of the IEEE/RSJ/GI International Conference on Intelligent Robots and Systems*, Munich, 1994, pp. 776-783.

[9] F. Mondada, E. Franzi, and P. Ienne, "Mobile Robot Miniaturization: A Tool for Investigation in Control Algorithms," in *Proceedings of the 3rd International Symposium on Experimental Robotics Experimental Robotics III*, Japan, 1993, pp. 501-513.

[10] K. Yeonhoon, K. Soo Hyun, and K. Yoon Keun, "Dynamic Analysis of a Nonholonomic Two-Wheeled Inverted Pendulum Robot," *Journal of Intelligent and Robotic Systems,* vol. 44, pp. 25-46, Sep. 2005.

[11] F. Grasser, A. D'Arrigo, S. Colombi, and A. C. Rufer, "JOE: A Mobile Inverted Pendulum," *IEEE Transactions on Industrial Electronics,* vol. 49, pp. 107-114, Feb. 2002.

[12] G. A. Bekey, *Autonomous Robots : From Biological Inspiration to Implementation and Control* The MIT Press, 2005.

[13] M. Railbert, K. Blankespoor, G. Nelson, and R. Playter, "BigDog, the Rough-Terrain Quadruped Robot," *The International Federation of Automatic Control Proceedings of the 17th World Congress,* pp. 10822-10825, Jul. 6-11 2008.

[14] S. Nabulsi, H. Montes, and M. Armada, "ROBOCLIMBER: Control System Architecture," in *Climbing and Walking Robots*: Springer Berlin Heidelberg, 2005, pp. 943-952.

[15] G. C. Haynes, A. Khripin, G. Lynch, J. Amory, A. Saunders, A. A. Rizzi, and D. E. Koditschek, "Rapid Pole Climbing with a Quadrupedal Robot," in *IEEE International Conference on Robotics and Automation*, Kobe, Japan, 2009, pp. 2767-2772.

[16] R. B. McGhee and A. A. Frank, "On the Stability Properties of Quadruped Creeping Gaits," *Mathematical Biosciences,* vol. 3, pp. 331-351, Aug. 1968.

[17] S. H. Hyon and T. Mita, "Development of a Biologically Inspired Hopping Robot-"Kenken"," in *IEEE International Conference on Robotics and Automation*, Washington DC, USA, 2002, pp. 3984-3991.

[18] Y. Sakagami, R. Watanabe, C. Aoyama, S. Matsunaga, N. Higaki, and K. Fujimura, "The Intelligent ASIMO: System Overview and Integration," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Switzerland, 2002, pp. 2478-2483.

[19] R. Chatterjee, M. Nagai, and F. Matsuno, "Development of Modular Legged Robots: Study with Three-Legged Robot Modularity," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sendai, Japan, 2004, pp. 1450-1455.

[20] J. Z. Kolter, M. P. Rodgers, and A. Y. Ng, "A Control Architecture for Quadruped Locomotion over Rough Terrain," in *IEEE International Conference on Robotics and Automation*, California, USA, 2008, pp. 811-818.

[21] G. Zhao, H. Zheng, J. Wang, and T. Li, "Petri-Net-Based Coordination Motion Control for Legged Robot," in *IEEE International Conference on Systems, Man and Cybernetics*, Washington DC, USA, 2003, pp. 581-586.

[22] M. R. Fielding, R. Dunlop, and C. J. Damaren, "Hamlet: Force/Position Controlled Hexapod Walker - Design and Systems," in *IEEE International Conference on Control Applications*, Mexico, 2001, pp. 984-989.

[23] S. Galt and B. L. Luk, "Predictive Terrain Contour Mapping for a Legged Robot," in *Fifth International Conference on Artificial Neural Networks*, Cambridge, United Kingdom, 1997, pp. 129-133.

[24] Y. Ogura, H. Aikawa, K. Shimomura, A. Morishima, L. Hun-ok, and A. Takanishi, "Development of a new Humanoid Robot WABIAN-2," in *IEEE*

International Conference on Robotics and Automation, Orlando, Florida, 2006, pp. 76-81.

[25] S. Sugano and Y. Shirai, "Robot Design and Environment Design - Waseda Robot-House Project," in *International Joint Conference SICE-ICASE* Busan, Korea, 2006, pp. I-31-I-34.

[26] T. Matsumoto, A. Konno, L. Gou, and M. Uchiyama, "A Humanoid Robot that Breaks Wooden Boards Applying Impulsive Force," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, 2006, pp. 5919-5924.

[27] K. Hosoda, T. Takuma, A. Nakamoto, and S. Hayashi, "Biped Robot Design powered by Antagonistic Pneumatic Actuators for Multi-Modal Locomotion," *Robotics and Autonomous Systems,* vol. 56, pp. 46-53, Jan 2008.

[28] C. Youngjin, Y. Bum-Jae, and O. Sang-Rok, "On the Stability of Indirect ZMP Controller for Biped Robot Systems," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sendai, Japan, 2004, pp. 1966-1971.

[29] T. McGeer, "Passive Dynamic Walking," *International Journal of Robotics Research,* vol. 9, pp. 62-82, April 1990.

[30] S. Collins, A. Ruina, R. Tedrake, and M. Wisse, "Efficient Bipedal Robots Based on Passive-Dynamic Walkers," *Science,* vol. 307, pp. 1082-1085, Feb. 18 2005.

[31] M. Wisse, G. Feliksdal, J. Van Frankkenhuyzen, and B. Moyer, "Passive-Based Walking Robot," *IEEE Robotics & Automation Magazine,* vol. 14, pp. 52-62, June 2007.

[32] A. D. Kuo, "Choosing Your Steps Carefully," *IEEE Robotics & Automation Magazine,* vol. 14, pp. 18-29, June 2007.

[33] T. McGeer, "Passive Walking with Knees," in *IEEE International Conference on Robotics and Automation*, Ohio, USA, 1990, pp. 1640-1645.

[34] P. Hoonsuwan, S. Sillapaphiromsuk, K. Sukvichai, and A. Fish, "Designing a Stable Humanoid Robot Trajectory using a Real Human Motion," in *6th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, Bangkok, Thailand, 2009, pp. 336-339.

[35] T. Takubo, Y. Imada, K. Ohara, Y. Mae, and T. Arai, "Rough Terrain Walking for Bipedal Robot by using ZMP Criteria Map," in *IEEE International Conference on Robotics and Automation* Toyonaka, Japan, 2009, pp. 788-793.

[36] K. Kaneko, F. Kanehiro, S. Kajita, H. Hirukawa, T. Kawasaki, M. Hirata, K. Akachi, and T. Isozumi, "Humanoid Robot HRP-2," in *IEEE International Conference on Robotics and Automation*, Japan, 2004, pp. 1083-1090.

[37] J. Watson, R. Ritzmann, S. Zill, and A. Pollack, "Control of Obstacle Climbing in the Cockroach, Blaberus Discoidalis. I. Kinematics," *Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology,* vol. 188, pp. 39-53, Feb 2002.

[38] E. Celaya and J. M. Porta, "Control of a Six-Legged Robot Walking on Abrupt Terrain," in *IEEE International Conference on Robotics and Automation*, Minnesota, USA, 1996, pp. 2731-2736 vol.3.

[39] R. Altendorfer, N. Moore, H. Komsuoglu, M. Buehler, H. B. Brown, D. McMordie, U. Saranli, R. Full, and D. E. Koditschek, "RHex: A Biologically Inspired Hexapod Runner," *Autonomous Robots,* vol. 11, pp. 207-213, Nov. 2001.

[40] G. M. J. Brazell, B. MacLaren, and W. W., "Mobile Work Platform for Initial Lunar Base Construction," in *2nd Conference on Lunar Bases and Space Activities of the 21st Century*, Texas, USA, 1992, pp. 633-635.

[41] D. M. Lyons and K. Pamnany, "Rotational Legged Locomotion," in *IEEE 12th International Conference on Advanced Robotics*, Washington, USA, 2005, pp. 223-228.

[42] I. Morazzani, D. Hong, D. Lahr, and P. Ren, "Novel Tripedal Mobile Robot and Considerations for Gait Planning Strategies Based on Kinematics," in *Recent Progress in Robotics: Viable Robotic Service to Human*. vol. 370 Jeju, Korea: Springer Berlin Heidelberg, 2008, pp. 35-48.

[43] G. Yanto, Y. XiaoLei, and A. Bowling, "A Navigable Six-Legged Robot Platform," in *IEEE International Conference on Robotics and Automation* New Orleans, USA, 2004, pp. 5105-5110.

[44] D. K. Pai, R. A. Barman, and S. K. Ralph, "Platonic Beasts: Spherically Symmetric Multilimbed Robots," *Autonomous Robots,* vol. 2, pp. 191-201, 1995.

[45] D. K. Pai, R. A. Barman, and S. K. Ralph, "Platonic Beasts: A New Family of Multilimbed Robots," in *IEEE International Conference on Robotics and Automation*, San Diego, USA, 1994, pp. 1019-1026.

[46] J. R. Rebula, P. D. Neuhaus, B. V. Bonnlander, M. J. Johnson, and J. E. Pratt, "A Controller for the LittleDog Quadruped Walking on Rough Terrain," in *IEEE International Conference on Robotics and Automation*, Roma, Italy, 2007, pp. 1467-1473.

[47] A. M. Hoover, E. Steltz, and R. S. Fearing, "RoACH: An Autonomous 2.4g Crawling Hexapod Robot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nice, France, 2008, pp. 26-33.

[48] W. Neubauer, "A Spider-Like Robot that Climbs Vertically in Ducts or Pipes," in *IEEE/RSJ/GI International Conference on Intelligent Robots and Systems 'Advanced Robotic Systems and the Real World'*, Munich, Germany, 1994, pp. 1178-1185.

[49] J. C. Grieco, M. Prieto, M. Armada, and P. Gonzalez de Santos, "A Six-Legged Climbing Robot for High Payloads," in *IEEE International Conference on Control Applications*, Trieste, Italy, 1998, pp. 446-450.

[50] D. Longo and G. Muscato, "A Modular Approach for the Design of the Alicia Climbing Robot for Industrial Inspection," *Industrial Robot: An International Journal,* vol. 31, pp. 148-158, 2004.

[51] D. A. Read, F. S. Hover, and M. S. Triantafyllou, "Forces on Oscillating Foils for Propulsion and Maneuvering," *Journal of Fluids and Structures,* vol. 17, pp. 163-183, Jan. 2003.

[52] T. Maneewarn and B. Maneechai, "Design of Pipe Crawling Gaits for a Snake Robot," in *IEEE International Conference on Robotics and Biomimetics*, Bangkok, Thailand, 2009, pp. 1-6.

[53] S. Hirose and P. Cave, *Biologically Inspired Robots: Snake-Like Locomotors and Manipulators*. New York, USA: Oxford University Press, 1993.

[54] A. Rezaei, Y. Shekofteh, M. Kamrani, A. Fallah, and F. Barazandeh, "Design and Control of a Snake Robot According to Snake Anatomy," in *International Conference on Computer and Communication Engineering*, Kuala Lumpur, Malaysia, 2008, pp. 191-194.

[55] J. Borenstein and A. Borrell, "The OmniTread OT-4 Serpentine Robot," in *IEEE International Conference on Robotics and Automation*, California, USA, 2008, pp. 1766-1767.

[56] H. B. Brown, M. Schwerin, E. Shammas, and H. Choset, "Design and Control of a Second-Generation Hyper-Redundant Mechanism," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Diego, USA, 2007, pp. 2603-2608.

[57] J. Ute and K. Ono, "Fast and Efficient Locomotion of a Snake Robot Based on Self-Excitation Principle," in *7th International Workshop on Advanced Motion Control*, Maribor, Slovenia, 2002, pp. 532-539.

[58] G. Junyao, G. Xueshan, Z. Wei, Z. Jianguo, and W. Boyu, "Design and Research of a New Structure Rescue Snake Robot with All Body Drive System," in *IEEE*

International Conference on Mechatronics and Automation, Takamatsu, Japan, 2008, pp. 119-124.

[59] M. Raibert, H. , "Legged Robots," *Communications of the ACM,* vol. 29, pp. 499-514, 1986.

[60] P. Fiorini, S. Hayati, M. Heverly, and J. Gensler, "A Hopping Robot for Planetary Exploration," in *IEEE Aerospace Conference*, Aspen, USA, 1999, pp. 153-158.

[61] R. A. Luders, D. Apostolopoulos, and D. Wettergreen, "Control Strategies for a Multi-Legged Hopping Robot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nice, France, 2008, pp. 1519-1524.

[62] X. Liu, M. Zhang, and W. Liu, "Methods to Modular Robot Design," in *Second International Symposium on Intelligent Information Technology Applications*, Shanghai, China, 2008, pp. 663-668.

[63] E. Østergaard, K. Kassow, R. Beck, and H. Lund, "Design of the ATRON Lattice-Based Self-Reconfigurable Robot," *Autonomous Robots,* vol. 21, pp. 165-183, 2006.

[64] R. F. M. Garcia, A. Lyder, D. J. Christensen, and K. Stoy, "Reusable Electronics and Adaptable Communication as Implemented in the Odin Modular Robot," in *IEEE International Conference on Robotics and Automation*, Kobe, Japan, 2009, pp. 1152-1158.

[65] S. Murata, E. Yoshida, A. Kamimura, H. Kurokawa, K. Tomita, and S. Kokaji, "M-TRAN: Self-Reconfigurable Modular Robotic System," *IEEE/ASME Transactions on Mechatronics,* vol. 7, pp. 431-441, Dec. 2002.

[66] A. Lyder, R. Garcia, and K. Stoy, "Mechanical Design of Odin, An Extendable Heterogeneous Deformable Modular Robot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nice, France, 2008, pp. 883-888.

[67] M. Yim, D. G. Duff, and K. D. Roufas, "PolyBot: A Modular Reconfigurable Robot," in *IEEE International Conference on Robotics and Automation*, San Francisco, USA, 2000, pp. 514-520 vol.1.

[68] H. Kurokawa, A. Kamimura, E. Yoshida, K. Tomita, S. Kokaji, and S. Murata, "M-TRAN II: Metamorphosis from a Four-Legged Walker to a Caterpillar," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Las Vegas, USA, 2003, pp. 2454-2459.

[69] L. Woo Ho and A. C. Sanderson, "Dynamic Rolling Locomotion and Control of Modular Robots," *IEEE Transactions on Robotics and Automation,* vol. 18, pp. 32-41, Feb. 2002.

[70] D. Christensen, D. Brandt, K. Stoy, and U. P. Schultz, "A Unified Simulator for Self-Reconfigurable Robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nice, France, 2008, pp. 870-876.

[71] Advanced Industrial Science and Technology Institute, "M-TRAN Publications List." Internet: http://unit.aist.go.jp/is/frrg/dsysd/mtran3/publication.htm, Aug. 5, 2008 [Jan. 25, 2009].

[72] K. Haruhisa, T. Kohji, K. Akiya, K. Shigeru, H. Takashi, and M. Satoshi, "Distributed Self-Reconfiguration of M-TRAN III Modular Robotic System," *International Journal of Robotics Research,* vol. 27, pp. 373-386, March 2008.

[73] A. Halme, I. Leppanen, M. Montonen, and S. Ylonen, "Robot Motion by Simultaneously Wheel and Leg Propulsion," in *4th International Conference on Climbing and Walking Robots* Karlsruhe Germany: CLAWAR, 2001.

[74] N. Eiji and N. Sei, "Leg-Wheel Robot: A Futuristic Mobile Platform for Forestry Industry," in *IEEE/Tsukuba International Workshop on Advanced Robotics, 'Can Robots Contribute to Preventing Environmental Deterioration?'*, Tsukuba, Japan, 1993, pp. 109-112.

[75] A. S. Boxerbaum, J. Oro, and R. D. Quinn, "Introducing DAGSI Whegs: The Latest Generation of Whegs Robots, Featuring a Passive-Compliant Body Joint," in *IEEE International Conference on Robotics and Automation*, Pasadena, USA, 2008, pp. 1783-1784.

[76] M. Eich, F. Grimminger, and F. Kirchner, "Proprioceptive Control of a Hybrid Legged-Wheeled Robot," in *IEEE International Conference on Robotics and Biomimetics*, Bangkok, Thailand, 2009, pp. 774-779.

[77] D. Xingguang, H. Qiang, R. Nasir, L. Junchen, and L. Jingtao, "MOBIT, A Small Wheel - Track - Leg Mobile Robot," in *The Sixth World Congress on Intelligent Control and Automation*, Dalian, China, 2006, pp. 9159-9163.

[78] K. A. Daltorio, A. D. Horchler, S. Gorb, R. E. Ritzmann, and R. D. Quinn, "A Small Wall-Walking Robot with Compliant, Adhesive Feet," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Alberta, Canada, 2005, pp. 3648-3653.

[79] K. A. Daltorio, T. C. Witushynsky, G. D. Wile, L. R. Palmer, A. A. Malek, M. R. Ahmad, L. Southard, S. N. Gorb, R. E. Ritzmann, and R. D. Quinn, "A Body Joint Improves Vertical to Horizontal Transitions of a Wall-Climbing Robot," in *IEEE International Conference on Robotics and Automation*, Pasadena, USA, 2008, pp. 3046-3051.

[80] M. Heverly, "A Wheel-On-Limb Rover for Lunar Operations," in *9th International Symposium on Artificial Intelligence, Robotics and Automation for Space* Los Angeles, USA: Pasadena, CA : Jet Propulsion Laboratory, National Aeronautics and Space Administration, 2008.

[81] K. Hauser, T. Bretl, J.-C. Latombe, and B. Wilcox, "Motion Planning for a Six-Legged Lunar Robot," in *Algorithmic Foundation of Robotics VII*. vol. 47: Springer Berlin Heidelberg, 2008, pp. 301-316.

[82] Q. Jia, Y. Zheng, H. Sun, H. Cao, and H. Li, "Motion Control of a Novel Spherical Robot Equipped with a Flywheel," in *IEEE International Conference on Information and Automation*, Zhuhai, China, 2009, pp. 893-898.

[83] S. Guanghui, Z. Qiang, and C. Yao, "Motion Control of Spherical Robot Based on Conservation of Angular Momentum," in *International Conference on Mechatronics and Automation*, Jilin, China, 2009, pp. 599-604.

[84] Z. Qiang, Z. Tingzhi, C. Ming, and C. Sanlong, "Dynamic Trajectory Planning of a Spherical Mobile Robot," in *IEEE Conference on Robotics, Automation and Mechatronics*, Bangkok, Thailand, 2006, pp. 1-6.

[85] R. Mukherjee and M. A. Minor, "A Simple Motion Planner for a Spherical Mobile Robot," in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Atlanta, USA, 1999, pp. 896-901.

[86] J. QingXuan, S. HanXv, and L. DaLiang, "Analysis of Actuation for a Spherical Robot," in *IEEE Conference on Robotics, Automation and Mechatronics*, Chengdu, China, 2008, pp. 266-271.

[87] F. Michaud and S. Caron, "Roball, The Rolling Robot," *Autonomous Robots,* vol. 12, pp. 211-222, 2002.

[88] GameStationX, "Sony Playstation Remote Control Command Structure." Internet: http://www.gamesx.com/controldata/psxcont/psxcont.htm, Aug. 13, 1998 [Nov. 5, 2009].

[89] Chipcon, "CC2431 Data Sheet." Internet: http://focus.ti.com/lit/ds/symlink/cc2431.pdf, Jun. 15, 2007 [Feb. 17, 2009].

[90] J. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*: MIT Press, 1992.

POSTGRADUATE FORMS
UNIVERSITY POSTGRADUATE BOARD

FORM – PG6 DEPOSIT OF MASTER'S THESIS / EXEGESIS / DISSERTATION IN THE AUT LIBRARY

This form is to be printed on acid-free paper. The completed and signed form should be bound into the copy of the thesis/exegesis/dissertation intended for the AUT University Library, i.e. the copy which is printed on acid-free paper. If the work is to be treated as confidential or is embargoed for a specified time, form PG18 must also be completed and bound into the thesis/exegesis/dissertation. For more information consult the AUT University Postgraduate Handbook.

Student's Name	Christopher Keith Coyte	**Student ID No**	0304888
Degree	Master of Engineering	**Year of submission (for examination)**	2010

Thesis ☒ **Dissertation** □ **Exegesis** □ **Points Value** 120

Title Spike: A Novel Cube-Based Robotic Platform

DECLARATION

I hereby deposit a print and digital copy of my thesis/exegesis with the Auckland University of Technology Library. I confirm that any changes required by the examiners have been carried out to the satisfaction of my primary supervisor and that the content of the digital copy corresponds exactly to the content of the print copy in its entirety.

This thesis/exegesis is my own work and, to the best of my knowledge and belief, it contains:

- no material previously published or written by another person (except where explicitly defined in the acknowledgements);
- no material which to a substantial extent has been submitted for the award of any other degree or diploma of a university or other institution of higher learning.

CONDITIONS OF USE

From the date of deposit of this thesis/exegesis/dissertation or the cessation of any approved access restrictions, the conditions of use are as follows:

1. This thesis/exegesis/dissertation may be consulted for the purposes of private study or research provided that:
   (i) appropriate acknowledgement is made of its use;
   (ii) my permission is obtained before any material contained in it is published.

2. The digital copy may be made available via the Internet by the AUT University Library in downloadable, read-only format with unrestricted access, in the interests of open access to research information.

3. In accordance with Section 56 of the Copyright Act 1994, the AUT University Library may make a copy of this thesis/exegesis/dissertation for supply to the collection of another prescribed library on request from that library.

THIRD PARTY COPYRIGHT STATEMENT

I have either used no substantial portions of third party copyright material, including charts, diagrams, graphs, photographs or maps, in my thesis/exegesis or I have obtained permission for such material to be made accessible worldwide via the Internet. If permission has not been obtained, I have asked/will ask the Library to remove the third party copyright material from the digital copy.

Student's Signature ______________________ **Date** ______________________