Knight Sweeper 4200 Autonomous Metal Detecting Robot

Phong Le, Josh Haley, Brandon Reeves, Jerard Jose

School of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida, 32816-2450

***Abstract* — This document discusses the design of our Knight Sweeper vehicle. Knight Sweeper is an autonomous vehicle designed to detect any metallic objects and avoid obstacles during its route of navigation. Once detection occurs the vehicle is designed to identify region of detection and avoid collision with the actual metallic object or obstacle in its way. Knight Sweeper will be able to operate in a manual mode as well as in autonomous mode. The project itself uses various hardware components to achieve its overall task. These components are GPS module, compass module, serial camera, wireless RF module, ultrasonic sensors, and metal detecting circuit.**

***Index Terms* — Autonomous, IED detector, obstacle avoidance, metal detector.**

I. Introduction

The Knight Sweeper can serve several useful purposes and applications. The technology that will be used in Knight Sweeper can be found in many common consumer electronics. The main purpose of Knight Sweeper is to automate a metal detecting vehicle that has an autonomous start to end route and search the specified parameter for metallic materials. This function can be quite useful for either military application or hobbyist's application. In the case of military application Knight Sweeper could be quite useful in finding metallic objects such as mines, traps, or improvised explosive devices so that a safe path can be paved. The main components of Knight Sweeper is a rover four wheeled platform, a microcontroller unit, ultrasonic devices, GPS unit, wireless module, and a metal detecting circuit.

The metal detector is composed of an integrated circuit with an internal oscillator. The external method of search for metal is done using a search coil that oscillates at a frequency which is close to that of the internal oscillation. A modern beat frequency oscillation method will then be utilized to determine detection of metallic objects. When detection occurs a bright LED will be lit, the rover will be flagged to stop its search path pinpoint the location of detection then run the algorithm to search for a new search path. Once detection occurs an analog output signal is sent to the main microcontroller unit to indicate that metal has been found. The microcontroller then records GPS location of detection and sends this information to the user interface which in return halts the actual rover vehicle for predetermined amount of time. Once the duration expires the pinpointed location will be removed from the vehicles search route and an alternate route will be determined and search will continue. The microcontroller also takes care of the controls of the rover vehicle through the utilization of a H-bridge design connector the a motor controller that controls vehicle acceleration and direction. Object avoidance will operate in a similar manner as the metal detection. Once the object is detected the microcontroller will be flagged and the rover will then cease to move until an alternate route is found.

The vehicle will be monitored on a personal computer through a custom GUI. From the interfacing of the personal computer the user will have the option to control the vehicle manually to override the autonomous mode. A visual view of the detection area as well as the pinpointed GPS location will also be available.

II. Requirements

The main requirements of Knight Sweeper are that it must be able to autonomously maneuver through a terrain powered by a 14.8 Li-Polymer Battery. During its route it must be able to detect IED's within a range of 4cm and also be able to avoid any obstacles within its course. Knight Sweeper must weigh no more than 6 lbs. Knight Sweeper must be able to communicate and send data to the user via telemetry. The final Requirements of Knight Sweeper is that it should be able to operate for more than one hour on a fully charged battery and have a top speed no less than 1mph.

III. Design Methods

A major goal in our design methods was to divide the project into two major categories "Overall Hardware" and "Overall Software". Within these two categories a further breakdown was done consisting of all the major components being implemented into Knight Sweeper. Each component was assigned to an individual group member in order to ensure each part get's done. This proved to be very efficient and organized. Each individual responsible for each component would report his updates during our weekly meetings. During these updates suggestions and critiques were given. From here all of our initial design and testing was done on a breadboard. Once tested to our specified requirements each circuit was then transferred onto a peripheral board before fabricating the actual PCB.

IV. Project Management

The first step in managing the project was to set a comfortable overall budget to which we could submit to Work Central Florida for funding. When budgeting we decided to budget for spare parts in case anything happened during the actual build and testing of Knight Sweeper. This proved to be very helpful when encountering problems such as burnt out parts or having to include additional parts into our design. Management of the overall progress for the project was assigned to one group member. This group member was responsible for contacting the rest of the group to get progress updates on their assigned components. This was done to avoid confusion so that if any conflict was seen only one individual was accountable. Major decisions in hardware of software were discussed amongst the group and collectively decided. Each member was assigned a particular task that they were responsible for completing. Although each task was done with the help of the entire group, one individual was responsible for the overall completion. Weekly meetings were held in order to discuss current design ideas, potential problems, and updates on progression. These meetings were vital in given the group a solid foundation on the current state of the project. The weekly meetings also proved to be very motivating such that if one person was further ahead than the rest of the group the other members would realize this and pick up the pace.

V. Implementation

Each circuit was tested to ensure that the components worked before the initial stages of implementation. This was done with the idea that more than often just because something works by itself doesn't necessarily mean that it will work once implemented with all of the components. We then checked the continuity of our printed circuit boards before soldering the parts needed. Once soldered and checked the PCB's were mounted and each individual component was checked to ensure that they operated to specified requirements.

VI. Metal Detector

When designing our metal detection circuit we wanted to go with a metal detection method that had an output which could easily be picked up by our MCU. This is why the design utilized a modern Beat Frequency Oscillation method. During BFO detection a sound output isn't necessary. BFO detection of metal is signaled by a change in frequency and a change in voltage which can visually be seen by utilizing transistors and LED's. The only issue of BFO versus other detection methods is the range given. The detection range sensitivity is changed by tuning a 1k potentiometer and 10k potentiometer which is part of a circuit designed with the IC TDA 0161. TDA0161 is an IC which acts as a BFO detector. The actual search coil was designed using magnet wire which was simply wound and tested by an LCR meter to have an inductance value that was calculated based on the values of the metal detecting circuit. When the search coil comes in contact range of metal the output pin is 1 volt. This

1 volt is then translated to the MCU. The IED detector circuit merely takes in an input in the form of an analog signal via an analog to digital converter. This module will contain three main functions to facilitate the needed uses of the IED detection circuit. The first function initializes the digital to analog input for use. This function will be called by main module during the system initialization phase. The second will manually read the analog to digital input value and return this to the calling code. This function is used for integration testing, as well as telemetry gathering. The last function for use is the comparator interrupt. The function pointer will be contained in the Stellaris nested vector interrupt table and will be forced to execute when the analog input is above a specified threshold. It will immediately stop the robot's motion as we found an IED. A flag will be set letting the AI navigation module know that an IED has been detected and it will use this flag to determine the next best step of execution.

VII. Control Hardware

The Knight Sweeper will be utilizing a H-bridge configuration IC. The use of an integrated circuit will prove to be very beneficial when saving space for configuration. The Texas Instruments model DRV 8432 is a new model with many advanced features. The device has a high efficiency and continuous current at 7 A which is more than adequate for the motors. The device can support pulse wave modulation control from the microcontroller being used. The integrated self-protection circuits of under voltage, over temperature, overload and short circuit make it an excellent option that does not have to be implemented outside the motor controller. The H-bridge configuration that will be used with our motor controller interface is produced by STMicroelectronic, model L298N. The L298N is a very commonly used motor controller which can support pulse wave modulation control. It has the ability to handle a wide range of operating supply voltages and our maximum supply current. The design has standard pin packages for schematic design and printed circuit board design for testing and simulations unlike the Texas Instruments model. According to the specifications it can support the design of controlling the left two motors and the right two motors with the two full bridges. The only drawback for this design is that it does not have internal diodes in the motor controller which will require external diodes and according to the data sheet Schottky diodes are recommended. The design with additional diodes is simple enough so the L298N is the selected motor controller for this design.

VIII. Automation

Knight Sweeper will implement autonomous mode through the implementation of the A* Algorithm. A* assigns each node a heuristic value that is equal to the cost to getting to that node, as well as the estimated cost to reach the objective. For us these costs are given as the Manhattan distance, or level one norm, which has been shown to be an optimal heuristic for the A* algorithm. The reason that the A* algorithm doesn't suffer from the same local minima issues that a greedy algorithm does is due to the heuristic taking cost of path traveled into account as well as the best looking next option. The procedure for this algorithm is as follows: 1. Check for the lowest value node in the open set. 2. Apply an operator to the node generating all children nodes based on possible movement from a node. 3. Check all children of this node that are not in the closed set for the destination else place them onto the open set and the current node onto the closed set. 4. Go back to 1. The use of a closed set prevents the algorithm from exploring cycles in the graph. By generating children of nodes as we explore the search space rather than generating the whole graph initially reduces the memory required from a possible infinite number of states down to an amount that is hopefully a constant multiple of the units between the start and end point. The artificial intelligence module is driving and controlling element of the robot that takes in data from its various sensor systems and then commands the motors to drive the robot to a new location. This is done using a polling routine that constantly assesses the state of the robot, and then calls for the robot to move. This module allows for the robot to be in several modes: Standby, AI Navigation, or Manual. In standby the robot is not doing anything, but merely waiting for a command. In manual the robot is waiting for manual locomotion commands to drive the robot. The automatic mode is the one that is most complex as it contains several steps and algorithms. AI navigation and manual mode can only be commanded from standby mode which can be commanded from any mode. The first thing that the embedded software does in automatic mode is take the given objective location, and the current location, and create a coordinate grid. This coordinate grid is then broken down into a graph that indicates all locations that the robot can occupy with the vertices between each point indicating an available path of travel. This graph is generated by applying all of the movement operators to grid locations and determining which locations are immediately accessible in a single move operation by the robot. Each node is assigned a heuristic value used in the actual search. Nodes in the graph that represent location of known obstructions or suspected IED's are removed so that a path is not navigated through one of these locations. An example is shown below where the only node generating operators available are movement to non-diagonal adjacent nodes and repeated nodes have been omitted.

IX. Control Software

Knight Sweeper will use a microcontroller that has interrupt capabilities which is desired as it will require less speed, and thus power, than a similar system using polling techniques to keep up with data processing in near real time. The choice of a particular microcontroller needs to take into account the problem and solution at hand, the autonomous navigation using the A* algorithm. This algorithm requires the construction and searching of a graph of locations. Every time the Knight Sweeper system encounters an obstacle or detects a possible IED, this graph will need to be modified and searched again. The speed and performance of the selected microcontroller must be able to keep up with this task as well as perform the background low level hardware interfacing. Another concern is the RAM available to the microcontroller available for use in the A* algorithm. If a 16-bit system is assumed with a two dimensional integer coordinate system, an approximately eight operators using a hexagonal grid system, then a node in the A* algorithm occupies twenty bytes of memory and thus only fifty-one nodes can be stored in a single kilobyte of memory. With the possibility of a large search space, a microcontroller that can support many nodes needs to be selected. Additionally, the use of a C/C++ environment is desired due to its familiarity and increased productivity over that of assembly level programming. The GUI is more than just simply a single module or class, but rather a collection of modules and classes that define both the GUI forms and the backend controlling elements behind them. The telemetry data will be held in a single object, the debugging functionality will be implemented in discrete forms. The main form will bind together all elements and handle the autonomous task and display telemetry. The GUI module will have a main form that the user interacts with and will handle operational considerations. When any pictorial representation is clicked a new form will launch that will display the picture taken by the robot. The Main GUI also contains the a view of the current locational data taken from telemetry so that values will be given in addition to the pictorial representation displayed in progress area. An area is provided to input a desired end point and command the robot into its autonomous mode. Console Output allows for text based error and logging information to be available to the user. The debug section of the form will launch new windows to allow for the debugging of individual systems of the Knight

Sweeper robot. Each of the spawned debug modules will be custom tailored to each system undergoing testing, but will be similar in setup in that they will take in user input, package the message to be sent to the robot, and then show output based upon the results of the action. To facilitate the telemetry data input that needs to be available to all of the forms, ie the main form and all debugging forms, a singleton pattern was used. A singleton is an object that will only be instantiated once and is a preferable way to implement data that needs to have global availability. This object will contain all telemetry and data from the robot simplifying the interactions between the GUI forms and the message passing interface. This design is subject to modification as new requirements are found, but as of now will allow for developer debugging, system integration testing, and serve as the Knight Sweeper user interface.

X. Obstacle Avoidance

Knight Sweeper will be using the LV-MaxSonar-EZ0 ultrasonic sensor which operates at a voltage of 2-5 to 5V with recommended currents ranging from 2ma to 3mA. The sensor allows for reading at rate of 20-Hz or equivalently every 50mS. The MaxSonar-EZ0 provides both very short to long range detection with a sonar range from 6 inches to 254 inches with a 1 inch resolution. There must be at least 10ms between the end of the echo pulse and the beginning of a new trigger pulse. The sonic burst sent out by the sensor is a set of 8 bursts at 42 kHz. Once the return of the sound wave is detected, an echo pulse will be sent on the echo output pin that is proportional in width to the distance of the nearest detected object. The width of that pulse ranges from 100µs to 18ms, or is 36ms if the pulse does not return to the sensor then we can deduce there is nothing to be detected. With sound traveling 1 inch every 73.746µs, it will take at most 17.4 ms for the pulse to return. Therefore, the maximum time between triggers of the device is about 64ms. The ultra-sonic range finder selected uses a UART interface that will be piped through the extended UART interface. It requires a void function, “USInit” to initialize and read data from the actual ultra-sonic range finder. Another function, “getDistance”, will actually read the data from the sensors and return this information to the calling code module. Due to the relatively slow speed of the robot, this sensor can be polled for the distance to the target without worrying about the possibility of crashing as the polling rate will be high enough to prevent such a scenario. This unit will interact with the navigation module, the telemetry module, and the message parsing module.

XI. Power System

When designing the power system for Knight Sweeper we wanted to incorporate a supply of power that was rechargeable. Out of all the options for rechargeable batteries we felt that Lithium Polymer technology was perfect for the direction we wanted. Lithium Polymers are known to be very light, cheap, and reliable compared to other choices. The 14.8V Lithium Polymer battery will be regulated to 5V and 3.3V using switching regulators. We chose a battery that exceeded the maximum need of voltage with the idea that we will regulate the voltage to the necessary specifications for each component. Using the voltage regulators we will be able to provide sufficient voltage along with adequate run time due to the large capacity of the actual battery. The main voltage source will be a 14.8V Lithium Polymer Battery rated at 5500 mAh. The motors will received a filtered voltage directly from the battery; it is also important to note that instead of motors receiving 14.8V, they are actually receiving approximately 12 due to the 2-volt drop across the diodes. Switching regulator were chosen over linear regulator was due to the fact the due to the gained efficiency. During testing we noticed that the linear regulators produced a large amount of heat, which lead to switching regulator being the obvious choice. We decided to go with the TL2575 family of regulators produced by Texas Instruments, which are up to 88% efficient, use a simple 5 pin out design and produce negligible noise for our purposes. In the simple three-pin design, the first pin is for the input, pin two is the regulated output and pin three through 5 are ground, feedback and a manual ON/OFF. While this family of regulators requires for additional components (diodes, capacitors, inductors) the trade off of space vs. efficiency was well worth it. This regulator allows a wide range input voltages up to 60V, which is more than sufficient four, our design purposes. We also adopted the use of heat sinks on our two regulators even though through testing we did not notice a large amount of heat being dissipated. This family also offers internal frequency compensation, a fixed frequency oscillator, cycle-by-cycle current limiting, and thermal shutdown.

XII. GPS

Knight Sweeper will use a 20 Channel SR-92 GPS Engine Board due to its high accuracy which will ease the complexity of navigation in its all-in-one design will allow for easy integration and testing. An additional benefit of the all in one module design is that it can be mounted independently of a PCB in a location that minimizes the amount of noise. This module requires no additional hardware circuitry to interface with a microcontroller's UARTs. To interface with the GPS an interface will have to be programmed to communicate with the module, and provide data in a useable manner to the microcontroller and navigational AI. This GPS location data will be used to tag the location of suspected IED's and obstacles and will provide special telemetry to the PC software to allow us to track the robots progress. The GPS unit is used for localizing the robot, planning navigation, and for logging the location of suspected IED's. The GPS module is a UART device that will interface with the extended UART module. It requires a function, "GPSinit" to initialize the GPS for use. A function "hasLock" will indicate if the GPS currently has a locational lock and thus can be utilized for navigation. The final function, "getLoc" required will query the GPS and return a data type containing the location of the robot. This module will interact with the AI navigation, telemetry, and message parsing unit to allow for control of the robot.

XIII. Vehicle

The selected chassis for Knight Sweeper is a A4WD1 four wheeled vehicle. The chassis chosen has a wide base and larger diameter in the wheels it should be versatile enough to traverse the type of terrain this design is suited for. The four wheeled vehicle is very commonly used and also very cost efficient. The availability to place four motors will give an appropriate amount of power to handle the approximate weight of all the components and the printed circuit board. The four motors chosen to power the chassis for Knight Sweeper are four brushed GHM-02 DC motors. These GHM-02 brushed DC motors are good for their reduction ratio and the stall torque which allows for higher weight load. This section is geared towards giving an external overview of how the vehicle will be designed and built in order to perform its given objectives and requirements set in the prior sections. The vehicle itself requires that multiple inputs and outputs to and from the actual rover platform. The diagram provided below provides a very generalized overview of these inputs and outputs. On the vehicle the hardware components will be programmed through a computer connection port. The antenna on the vehicle allows for detection of position from GPS satellite broadcasts. Operation of the vehicle will be battery powered and a power port will allow for recharging of these batteries. The vehicle will also have the option of switching between autonomous or manual control. The overall design of the project has many objectives from the metal detection, obstacle avoidance sensors, microcontroller selection, GPS, wireless communication, autonomous control, manual control, motor controller type and the ability to navigate on desert like terrain. In order to accomplish these tasks the basic overall design of the hardware would need to be taken into account. Because of the number of components used in the project placement and orientation of them is a primary concern. The obstacle avoidance sensors and the metal detection circuits are the major components that would require orientation. It was also important to consider the potential interferences the hardware would cause each other. The Knight Sweeper will traverse in a forward direction the majority of the time it is used and so the metal detection will need to be in the front of the vehicle and be able to sense metal the entire width of the vehicle so that no part of it might detonate an improvised explosive device. The vehicle is likely to metal parts in its design and therefore must not interfere with the metal detecting sensor placed on the front; the metal detector itself must be far enough away from the chassis so not to get interference. The vehicle design or type will also need to be able to counter the weight of the sensor being placed on the front or be able to handle a counter weight placed inside. The obstacle avoidance sensors will also need to be pointing in the forward direction and angling in the right and left direction to account for potential obstacles when turning. The forward facing sensor will include a range finder in order to account for the appropriate breaking distance for the vehicle.

XIV. Microcontroller

The Stellaris M3 will be used as the MCU for the Knight Sweeper system. The deciding factor for the Stellaris M3 was the amount of memory available due to the fact that the graph theory algorithms needed for navigation are expected to create a number of nodes.

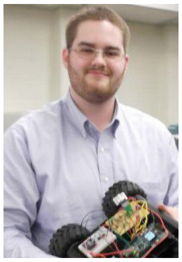
The Stellaris comes with a wealth of sample code in the form of the Stellaris Ware package to aid developers in using the various peripheral functions of the M3 processor. For the Knight sweeper system it was decided to use a development board provide by Texas Instruments as the controlling processor of the system and then just interface to a sensor breakout board. The development board provides power regulation for the processor as well as a number of integrated peripherals. The board provides JTAG emulation via a mini-USB port which allows for a unified coding, testing, and runtime environment. Additionally the board provides an organic light emitting diode display that will decrease the burden of debugging and software development.

It was observed that few MCU' had as many UARTS as is necessary to interface with all components and thus a solution had to be considered. To this end a solution was found to multiplex a serial signal using an analog multiplexer. This solution uses a 74HC4052 Serial/Analog Mux/Demux and is discussed further in the hardware design section of this document.

XV. Conclusion

The overall project proved to be very comprehensive with the number of hardware interfaces utilized, and complexity of the problems being solved to implement the desired functionality. We were able to implement and meet the specified requirements of Knight Sweeper. Overall IED detection was successfully done using BFO technology and obstacle avoidance was done using ultrasonic technology. The vehicle was successfully able to be controlled by both manual and autonomous mode with the functionality of using GPS to mark any location of metallic objects. Autonomous operation was achieved through the implementation of the A* Algorithm. Overall Knight Sweeper as a whole was a success and proved to be very rewarding.

Knight Sweeper Group Members



Joshua Haley will be graduating from the University of Central Florida Summa Cum Laude with a Bachelor's of Science of Physical Engineering majoring in Computer Engineering, minoring in Computer Science and Mathematics. In addition he is graduating with University Honors, participated in the Lockheed Martin College Work Experience program and completed a Honors in the Major Thesis involving a swarm robotics. Following graduation Joshua will be joining Soar Technology as a Software Engineer.

Jerard Jose is a senior at the University of Central Florida. He plans to graduate with his Bachelor's of Science in Electrical Engineering in May of 2012. He is currently looking for work in a related field with emphasis on system power analysis and design. He is considering graduate studies in Electrical Engineering or a MBA.

Phong Le will be graduating the University of Central Florida with a bachelor's in Electrical Engineering. He is highly active within the UCF Greek community, being a member of Sigma Lambda Beta International Fraternity, Inc. for which he was president of the organization for two years. Phong has also taken advantage of the highly valuable programs that UCF has to offer such as the Lockheed Martin College Work Experience Program. Phong has been part of the Lockheed CWEP Program for about 3.5 years where he has gained very valuable experience. Following graduation Phong will be joining Texas Instruments to start his career.

Brandon Reeves will be graduating from the University of Central Florida with a Bachelor's of Science in Electrical Engineering, minoring in Mathematics. In addition to this, Brandon has worked as a tutor in mathematics for UCF and participated in undergraduate research since his sophomore year. He spent two years working in the Intelligent Systems Lab under the supervision of Dr. Gonzalez and one year researching with Dr. Choudhury in the Mathematics Department. As a result of his research he recently completed a publication in the area of chaos theory and nonlinear dynamical systems. He hopes to continue his studies by eventually pursing an MBA after gaining work experience. Upon graduation Brandon will be joining Texas Instruments to begin his career.