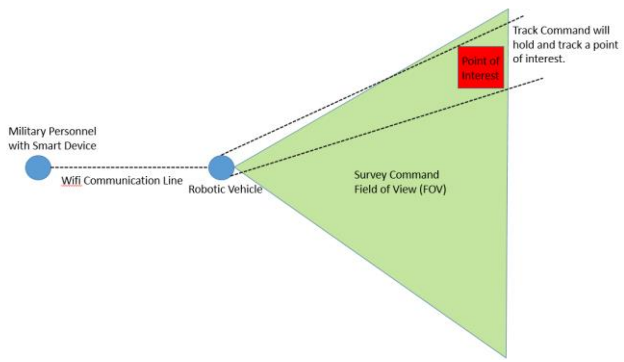
Military Surveillance Robotic Vehicle Title

Ryan Hromada, Adam Baumgartner, Austin King and Kevin Plaza.

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

Abstract **— Military personnel face many unique challenges on a daily basis in their work compared to civilians. Many of these challenges can be aided or overcome by having more information at the right times. Intelligence is generally gathered in many ways including covert operations, communication interception, interrogation, aerial surveillance, and ground surveillance. Our simple solution will help service this market by allowing for an autonomous robot to be given commands via a portable handheld device using only voice controls. This robotic vehicle will take in the voice command from the operator, and using its built in software, will carry out the command until the operator gives it a new command.**



Index Terms **— Ultrasonic transducers, Cameras, Autonomous systems, Wireless Communication.**

I. Introduction

Our project seeks to improve the solution of robotic ground surveillance by developing an autonomous robotic vehicle using computer vision and ultrasonic sensors that will aid the military personnel in finding a point of interest such as an injured individual, bomb, enemy combatant, vehicle, and much more. The robotic vehicle will have three primary functions to demonstrate how our robot could achieve this. The functions being survey for a point of interest (POI), track a POI, or find a POI. Surveying a POI means that the robot will move to the target location and use its camera to gain a visual of an entire area. This is a basic reconnaissance mode that is useful gaining information on an area and for finding a potential target. The "track" command will be used to follow a desired target. A color or special target will be designated as the tracked object, and the robot will be programmed to recognize the designation and once it is locked, it will follow the target and keep a visual of it in the camera. The "find" command will cause the robot to find a target with the designation. Along with the survey function, the robot can move to a desired location and search an area for a target, then with the track function the robot can lock onto that target and follow it until the command is canceled.

All of the mentioned functions would aid in ground reconnaissance by making it so that the robot can autonomously perform the reconnaissance without having a dedicated operator constantly controlling the robot. The controlling device would be simple to use, so that anyone could operate the robot, and the commands would allow for the user to simply give a command and the robot will automatically perform the task while freeing the operator to do another task while the robot performs its mission. These three functions mentioned will be activated from a voice command given into a smart phone or device by the military personnel.

Our motivation for choosing this project is that we would like to incorporate a project the includes equal elements of Computer Engineering and Electrical Engineering so that we can use the full spectrum of our degrees equally since we have two members from each discipline. We feel that his project offers unique challenges by designing smart algorithms that allow the user to interact with the vehicle in a much more unique fashion. Unlike some other robotics projects, our design is aimed at being a functional real world device that can be a useful tool for military or police.

Fig. 1 Functional Diagram

The robotic vehicle will be Wi-Fi enabled to allow for communication of the voice commands from the user to the vehicle for which mode to enter. This will also be used for the capability to provide camera feed back to the user. This will provide the user with the ability to see what is being surveyed, tracked, and/or found. The camera mentioned will also be used for the computer vision section of our project allowing the robot to distinguish between what is a POI and what is not. The last addition to the base robotic vehicle is ultrasonic sensors that will be positioned around the vehicle to allow it to run autonomously without collision.

II. Objectives

Below is a list of objectives that we have determined are a requirement for our vehicle to meet. Each of these objectives are will be explained in the following subsections.

A. Autonomous Operation with voice commands

One of the major components of our design is the ability for the robotic vehicle to be controlled through voice commands. This is desirable because it allows the operator to command the vehicle without the use of manual controls. This means that the controller can be a light and small device that leaves the operator free to perform other tasks while still having control of the vehicle. The vehicle must also be able to operate autonomously. The more that the vehicle can operate on its own, the less attention that the operator has to put into controlling it. The operator should be able to simply give a command and then the vehicle should be able to carry out the command without extra input from the user. The operator can then simply watch the display to gain the visual information without having to worry about steering the vehicle.

B. Operation in different modes

Our design will feature three modes of operation to carry out different types of tasks. These three modes will be "survey", "find" and "track". The function of these three modes will be implemented in the software of the vehicle with the use of a microcontroller. The external components such as the servos and camera shall be interfaced with the software to serve as inputs or outputs. For example, when a command of "track" is given through the controller device, the software shall recognize the keyword and execute a subroutine that reads the input from the camera to find the designation. Then the software shall output to the servos the signals required to move the vehicle in the direction of the target while reading the input from the ultrasonic sensors to avoid collisions. A similar procedure shall be executed for the other two modes to ensure correct operation.

C. Integrating multiple components onto a Printed Control Board (PCB)

From the requirements of the design project, we must incorporate a PCB into the design. With multiple sensors and servos utilized in our design, the PCB is required for us to combine all the electronic components together and provide power to the system. Some of the components can be directly interfaced with the microprocessor so they will not be included directly in the PCB design. The overall block diagram of the system is shown in Figure 2.

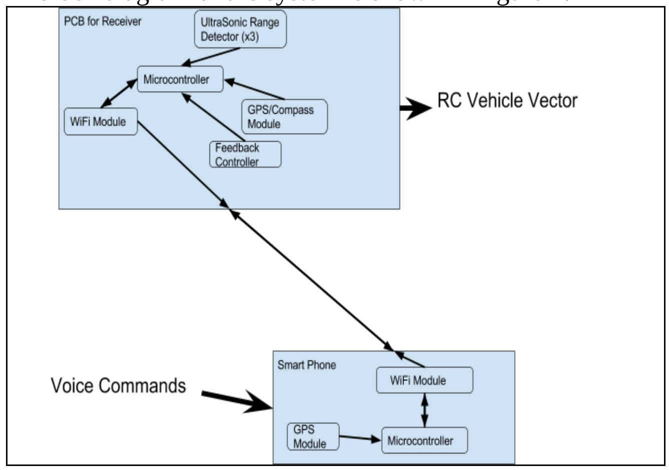


Figure 2. Block diagram

III. Computer Vision

The computer vision portion of the firmware needs to be able to identify a shape of a particular color. The key to implementing this is to define and mark the edges of the shape easiest of shapes to do this with will be a square or rectangular shape. Then using a specific color that is easily outlined in contrast to its surrounding environments such as a bold red. These parameters will make identifying the pre-determined point of interest easier to allow for focus on the portion of the algorithm that will affect servo and motor performance based on the visual feed and logic from the camera. SimpleCV, CV standing for computer vision is a program that provides an easy library for implementing computer vision on microcontrollers for project such as this one. The flowchart below shows the logic mentioned above before the data is used to control the Robotic Surveillance Vehicle.

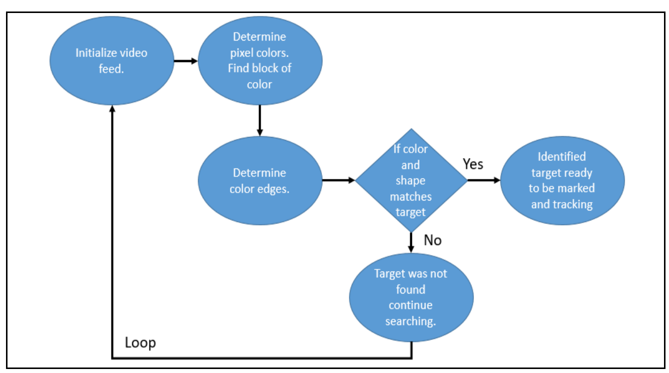


Figure 3. Computer Vision Algorithm to Determine a Target

Since the parameters for the point of interest has been defined for the benefit of the system now a look into how this will be implemented into the whole algorithm. During the survey mode the Robotic Surveillance Vehicle will scan an area of at least 120 degrees in front of the Robotic Surveillance Vehicle. During this time the Robotic Surveillance Vehicle will be attempting to identify a point of interest per the parameters mentioned above. Once the point of interest is identified it will be marked. Marking is defined as holding position on the point of interest keeping it in front of the Robotic Surveillance Vehicle and centered in the camera view to allow for tracking if initiated by the user via the smart device.

The Robotic Surveillance Vehicle will have multiple software components including the firmware onboard the Raspberry Pi microcontroller, and the mobile application software.

The firmware onboard the Raspberry Pi microcontroller will consist of the various sensors used to control the Robotic Surveillance Vehicle such as, the servo controller, ultrasonic sensors, camera, and digital compass. These will be used to implement the collision detection and avoidance as well as, computer vision algorithms to guide the Robotic Surveillance Vehicle as it tracks. Surveying and tracking points of interest will be the Robotic Surveillance Vehicle's primary function, the software algorithms mentioned above will be integral to these functions. These functions will be implemented on the Robotic Surveillance Vehicle, but will need to be integrated and communicate with the smart device via the mobile application used on it.

The software to be used outside of the Robotic Surveillance Vehicle will be handled through a mobile application on a smart device. The Robotic Surveillance Vehicle will be able to communicate with the mobile application via a UDP Wi-Fi connection. This will allow for voice commands to be transmitted from the smart device to the Robotic Surveillance Vehicle as well as, transmitting video and status updates of the sensors from the Robotic Surveillance Vehicle to the smart device.

IV. SENSOR ARRAY

A. Ultrasonic Transducer

In order to safely operate autonomously the vehicle needs the ability to actively avoid object detection. The ability requires some sort of sensor feedback to the MCU in order for the processor to determine whether the object is within range of collision with the vehicle. There are variety of methods for collision detection used in robots, however, the main methods are Infrared, Sonar, and Laser. Ideally to keep a lower power setup you would choose the method to best suit the project's goals. The key characteristics to judge the viability of the component for the project are the detecting distance of the component, the power consumption of the component and the. This means where the detection can start and where the detection can stop. Some devices are highly accurate at close ranges of around 1-2 cm whiles others can range up to 900 cm. Some of the advantage to using the Sonar is that the average angle for measurement is 30 degrees from the point of detection and minimal power consumption of around 15mA. The figure below shows that as distance increase the device becomes less precise with its readings and has a reduced margin of error.

Below shows a picture of the Ultrasonic detector connected to a breadboard used to test the effective range and width of detection. The picture is used for reference only and does not represent the actual testing implementation of the device. As you can see in the below picture the device has a reasonable degree of confidence up to 6 feet and a width of about 30 degrees. However, the further the object is from the device the less accurate the device will be if the object isn't directly in front of it.

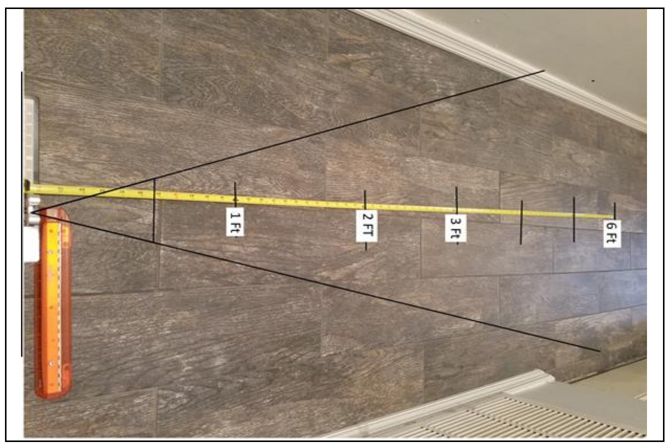


Figure 3: Example of viewing angle of ultrasonic sensor

When looking into the laser based applications most of the laser detection tends to be a narrow focused beam which eliminates the ability for the sensor to be stationary mounted. The power consumption for the systems tend to be higher when receiving data but have a low idle point. Since the device needs to be turned on and the angle of detection needs to be sufficiently wide the laser based sensor was not selected.

The infrared based sensor use much less power in comparison to the Ultrasonic or sensor based sensor and would not be susceptible to matching frequency interference. The infrared sensors also take up less over all space and are efficient at close range detection. These sensors would be ideal for our applications however their range is shorter in comparison to the ultrasonic sensor with long range sensors reaching up to 5 feet. The infrared sensor needs a minimum distance to function of 10-15 cm while the ultrasonic sensors can typically operate at a minimum distance of 2 cm.

After choosing the ultrasonic sensor for its better fit qualities for the project design the next important consideration is the method of communication to the MCU. The Ultrasonic sensors can come in a variety of output methods, Pulse Width Modulation, Analog and serial communication. Depending on the quality of the one or more of these options may be available. We chose a module the specified the Pulse Width Modulation method as this correlated with a lower cost component. The PWM works by initializing the unit through the sending of a 5-volt source for at least 10 µs. This triggers the unit to emit a series of pulses at a 40 kHz frequency. The pulses then bounce off an object and return to the receiver where they are echoed back to the MCU via 5-volt source for a period of time. The length of the time corresponds to the distance where the distance in centimeters is equal to the time output divide by 58. The time of the pulse is measured in a range of 150 µs – 25 ms. The formula is:

$$\frac{Time * 10^6}{58} = distance(cm)$$

The signal output is detailed below. The output response signal shows the length of the Pulse that is used to determine the distance of the object.

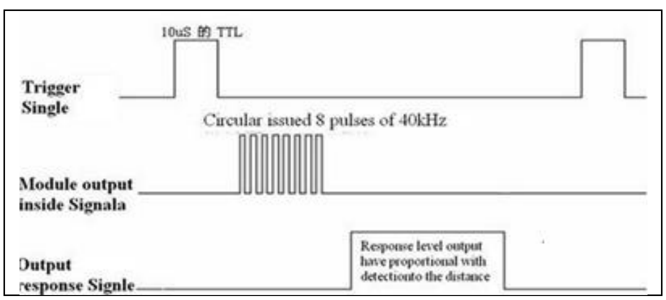


Figure 3-3: Timing diagram for ultrasonic sensor [1]

Based off the selections for vehicle collision avoidance we will need three sensors to accurately guide the vehicle through its directives while avoiding obstacles. By placing them on the sides and the front of the vehicle the vehicle will be able to determine an obstacle upcoming in its path.

Since we do not intend for the vehicle to work in a dynamic environment we intentionally left off a fourth detector in the back of the rear of the vehicle as the tests will be done in a controlled and static environment. The one important consideration is that the pulse needed to activate the sensor is 5 volts, found on most MCU's however the return pulse will be 5 volts so we will need a simple voltage divider to reduce the voltage as most MCU I/O pins operate at 3.3 volts or lower. When introducing this component to the PCB board we will need to ensure that this remains separate from the components that utilize the i2c protocol. As this device communicates exclusively with a basic PWM I will need to be connected to the GPIO line of the MCU.

B. Camera

When determining the camera choice for the autonomous robot design the choice was slightly more difficult than expected. While many devices make use of optical sensors and camera modules with varying pixels, it was difficult to find an industry standard when it came to MCU's. There some Display Serial Interface standards utilized by various manufacturers for Cameras one of which is the Camera Serial Interface (CSI) which is a specification of the Mobile Industry Processor Interface Alliance (MIPI). These were developed to define an interface between a camera and a host processor. The idea for the industry alliance was to allow the system integration of the interfaces to have less involved and make the processor easier and smoother for the companies involved. [2]

When it comes to the MCU's a few of the products have adopted the while some products have adopted the i2c serial interface to allow fewer input lines and faster communications. There are some modules that also utilize traditional UART interfaces allowing the user to customize the Baud rate based on communications protocols. Given that most of the cameras designed to interface with the various MCU's are sufficiently small we chose to go with the Raspberry Pi Camera for two reasons. The main reason is that the camera was designed to interface with our MCU choice. This allows for less debugging and less initialization of the camera based software. This particular camera meets a minimalist design and weight while having a robust interface following the MIPI Alliance standard. Our second is reason is that not only does it follow the MIPI standard is that it also follows the CSI specification. By using this specification, the camera can connect directly to the Raspberry PI GPU and bypass the CPU. This gives an increased performance as you can have the data sent straight to the GPU for encoding.

Due to the length of the cable we will most likely need to add extensions to the length. This will allow the camera to be mounted on a separate physical platform that can be raised above any jumper wires and subsystems of the RSV. The below picture illustrates the size of the camera relative the Chassis of the RSV. The overall size of the camera has to be small enough as to not interfere with the multiple subsystems

C. Magnetometer

When choosing the digital compass, the primary focus was to have the device to determine its current position relative to the magnetic north. The reason this information is needed is that the device can accurately determine where it went to help created a 2d map of travel. This along with an accelerometer can provide two methods to determine the amount the vehicle rotated. This will allow the vehicle to know the direction it has turned in to help determine its vector when plotting the map. The sensor arrays for the magnetometer has become limited in choices with most variations of the components are of higher accuracy or include extra sensor technology. At first our original thought was to only include a digital compass however many components are sold with triple-axis accelerometers and magnetometers combined.

We have decided on a lower cost solution that has an accuracy relevant for the project and the tasks that are needed to accomplish. The unit comes recalibrated, however, the component can be calibrated using the MCU or a traditional PC. This will help to determine axis offsets for both the accelerometer and magnetometer.

The device uses i2c communication protocol. The i2c communication protocol will reduce the wiring needed between the device and our MCU. Rather than having an input pin for each axis in each component this allows for serial communication between the MCU and the device via to serial ports. The most important about of our calibration is to get the component level on installation so that when the device reads a change in direction its true to our vehicle. Any deviations in the placement of the component could allow for incorrect position recordings and require extra calibration steps. The accelerometer and magnetometer will read out in the X, Y and X axis and allow for the reading of orientation of the device. In the absence of the local magnetic fields the device can read the Earth's magnetic field (20-60 micro Teslas) and determine the angle of the magnetic field relative to its position. This will allow us to utilize the X and Y axis for the compass component. [3]

D. Encoder

Rotary encoders measure the rotation of a shaft, either digitally or through an analog signal. The majority of encoders are either magnetic or optical. Optical encoders are cheap and rely on a photo sensor which measures visual changes on a disk rotating around the axis being measured. The principal drawback of this technology is a susceptibility to conditions that obscure visibility, such as dust. Magnetic encoders are robust, but they are more expensive than optical encoders. The proposed design is intended to work indoors. It is unlikely that dust or smoke would be a significant issue under these operating conditions. An optical sensor is sufficient for the proposed design and it would be superfluous to include a magnetic encoder. For these reasons an optical rotary encoder is the ideal choice for this design.

Encoders can be either absolute or incremental. The absolute rotary encoder measures the specific orientation of the shaft. The angle of rotation is determined by a series of concentric rings which each correspond to a binary value. All bits set to zero might represent the smallest positive angle and the binary number would count towards the largest measured angle. The caveat to this precision is that each ring requires a separate sensor and increases the design complexity. To justify the increased intricacy, there must be a palpable benefit to the angle information. In the current design the rotary encoder is used to determine information about an axle powering a vehicle. Wheels and tracks, the two proposed systems of propulsion, are uniform along all points of rotation. Both designs are not going to be significantly different at any orientation so there is little justification for recording angular information.

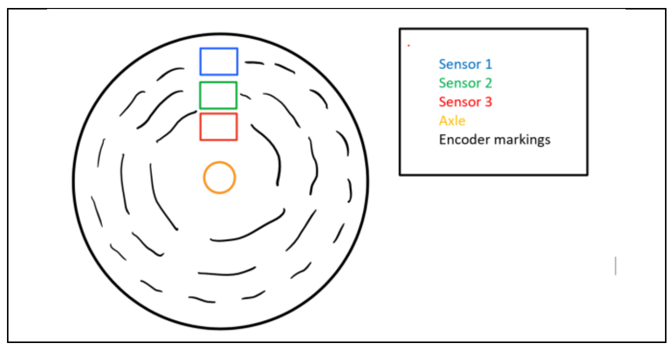


Figure 4: Absolute Rotary Encoder Diagram

An incremental encoder measures changes in shaft position without recording the exact location relative to a known value. A general design involves a sensor which detects changes which are evenly spaced around the axle. From the evenly spaced indicators the amount the shaft has rotated can be calculated. Since each increment is identical, it is impossible to know the precise angle of rotation. The benefit of an incremental encoder is that it requires one sensor which is considerably simpler than an absolute encoder. In this design, the rotary encoder is utilized to determine distances traveled. Each rotation can be interpreted as propelling the rover a set distance which should be consistent. The pertinent information is the number of rotations, not the absolute orientation of the axle.

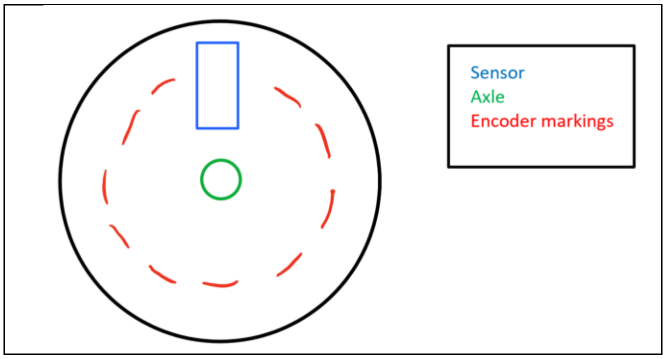


Figure 5: Incremental Rotary Encoder

Incremental rotary encoders can be further divided into standard or quadrature devices. The standard encoder records only changes in position with one sensor reading the increments around the axle. A quadrature encoder uses two staggered sensors that are ninety degrees out of phase. Depending on which sensor triggers first the direction of rotation can be registered. As the design relies on the encoder data to determine distances, it is plausible that the quadrature encoder is justified. Such devices would allow the rover to use reverse motion and record the data. A standard incremental encoder would not be able to distinguish between the two drive modes and would categorize both as forward motion.

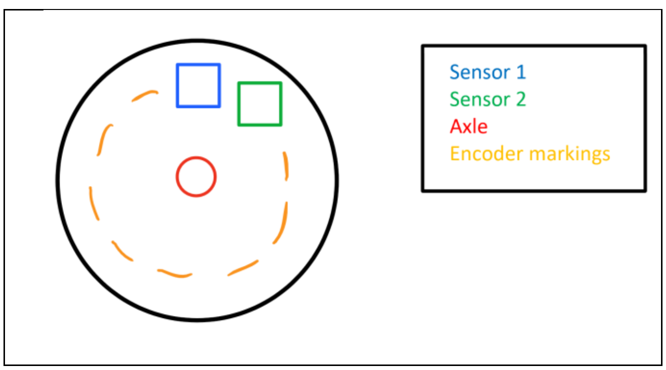


Figure 6: Quadrature Rotary Encoder Diagram

Though the extra data is potentially useful it is likely redundant. The rover is providing the propulsion directly through its engines and both the engine and encoder data are available. If the engine is operating in advancing mode, the encoder information should be interpreted as forward movement. If the engine is providing reverse motion the encoder data should be analyzed to indicate retreating distance. As this system accomplishes the same objectives as the more intricate quadrature sensor with a streamlined design, a standard incremental optical rotary encoder should be used.

The most prudent option was the Robot Shop RB-Rbo-122 encoder pair. These are a pair of low cost optical, incremental rotary encoders. The RB-Rbo-122 encoders contain all of the functionality required to implement this design, without excessive inclusions that would overcomplicate the robot. The encoders contain sixteen counts per rotation and it is not a quadrature design. Though not the most precise encoders available, they allow all of the precision required to implement the proposed rover.

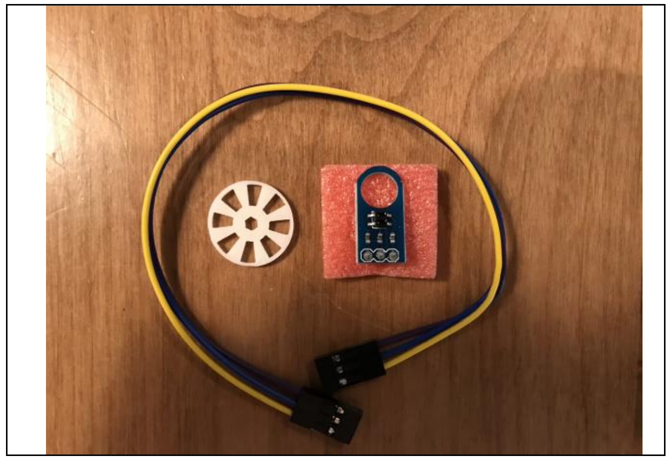


Figure 7: Optical Rotary Encoder

While the RB-Rbo-122 encoder is an attractive option for this design, it has a significant drawback. The output from this chip is analog and there are not any analog input pins on the Raspberry Pi. Though analog output without any available input is the primary drawback, it is not a significant obstacle. The inclusion of an analog to digital converter is an easily manageable addition to the current design. The Adafruit MCP3008 is an eight channel, ten bit analog to digital converter and enables the two optical encoders to be used with a Raspberry Pi.

V. Vehicle Movement

The servo selection was limited in that the servos needed to function in the tracked based chassis. We would need a motor for each track as the chassis operates each track separately. This allows for precise turning and is superior in pivoting over other concepts. Since we do not require the device to reach high speeds the overall speed limitation of this device can be negated.

When deciding the servos for the vehicle we had to major selections. We could go with a standard DC motor and no gearbox, or a DC motor with a gearbox. In either configuration we would need to ensure that the device can fit into the chassis and a rotary encoder can be installed to see the movement of the device.

The HS-322 pictured below can be used in the chassis design with some modifications done to the chassis. The advantage is that this a standard design and has good usage and review around the motor design to ensure that this device would be useable in our application and has a high top speed.

Figure 8: HS-322 [4]

The Pololu Size 130 Brushed DC motor shown in figure 9 draws only 70mA and operates anywhere from 3V to 9V. This motor is depicted below and meets all of our design specifications, though similar to the Tamiya design they are distinguished by a lighter plastic cap.

Figure 9: Pololu Size 130 Brushed DC motors

The device can be purchased with a gearbox with a twin motor configuration to make the connection to the tracked chassis much easier. The twin-motor gearbox made by the manufacturer of the FA-130 motors is a best fit for our vehicle as it offers a small concise package to house both of the of the motors. We chose this to power the device over the larger more powerful motor.

VI Multiple Slave and Master MCU

While researching the design of the device we've realized that our vehicle uses multiple components that require Pulse Width Modulation. This is used in the Ultrasonic Sensors, Rotary encoders and the Servo Driver. Due to this requirement we've implemented and Analog to Digital converter, a voltage divider and the General Purpose Input and Output pins of the MCU. After much research we've come across another solution that varies from the processes described above. In this solution we will utilize either a small MCU that can be programmed with the primary MCU. This device will be small in size, around the same number of pins as a typical Op-Amp but will be able to perform basic logic commands. The benefit to using the device is we will be able to combine some of the functionality of the Analog Digital Converter, the Servo Driver and the GPIO pins used in the sensor Array. The Servo driver is meant to be used with up to 16 different servos while the analog to digital converter is capable of 8 channels. Since this is far exceeding the needs of our device we have determine that we can replace these components plus the logic inherent in the GPIO to Ultrasonic Sensor Array with a Slave MCU.

The Slave and Master Concept isn't a new concept and has been utilized for some time primarily being credited to the Motorola Company. [5] This concept arose as a way to communicate with a Master controller and Slave controllers. The primary method of communication is the MOSI and MISO lines described below. Each of the slaves can be wired up in parallel with a Slave Select line enabling the device the Master MCU chooses to communicate with. The protocol utilizes Serial Synchronous Interface which means that the device will send data over sequentially one bit at a time. The advantage is that this means the devices can use less lines to communicate with. To ensure that the two devices are communicating properly and data is not lost the Master MCU will set a serial clock. The Slave MCU will communicate synchronously with the Master MCU via the two serial lines, MOST and MISO.

There are major benefits to using an MCU slave master design concept over the use of the traditional single MCU. One of the biggest benefits is the Voltage tolerance of the smaller MCU's. The voltage range for the I/O ports are typically 1.8 to 5.5 volts. This is of particular importance to the ultrasonic sensors as they will output in 5 volts and our Raspberry Pi can only handle a 3.3 volt input to the GPIO. The next advantage is some come with built in analog to digital converters, this can help us with our rotary encoders as we won't need a larger fully dedicated ADC. Since these slave type MCU's typically perform multiple functions it will allow us to utilize the ADC while still operating the devices that require PWM. The last beneficial ability is that the device can communicate i2c. The reason why this is important is that will reduce the need of the GPIO lines. It will allow the Raspberry pi communicate with all of our components via two lines.

The ATtiny chips can be programmed via the large microcontrollers including our choice of the raspberry pi. We will be able to specify any of the coding inside the smaller device so that it can act as an interface to the Sensor Array and Servo Controller. We can specify the i2c address so we won’t have any conflicts with the magnetometer. This will also allow us to run multiple ATtiny’s in parallel so that each can perform a different function.

Acknowledgement

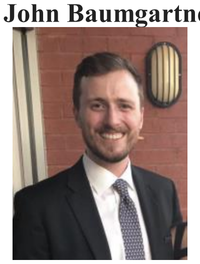
Kevin Plaza

Kevin is receiving his bachelor's in Computer Engineering from the University of Central Florida. He is presently working at Lockheed Martin Missiles & Fire Control in the College Work Experience Program. He plans on staying at Lockheed Martin MFC in a full time position as a Systems Engineer. He is currently a member of Tau Beta Pi and the UCF Honors College.

Austin King

Austin is a senior at the University of Central Florida studying Computer Engineering working part time for Pinnacle Solutions, Inc. as a Software Engineer looking towards full time employment with Pinnacle following graduation. Austin King plans to pursue a Masters in Computer Engineering or Computer Science in the coming years.

John Baumgartner



John is a UCF senior studying Electrical Engineering and will be attending law school at the University of Alabama. He hopes to use his engineering degree to practice as an intellectual property attorney.

Ryan Hromada

Ryan is receiving his bachelor’s in Electrical Engineering from the University of Central Florida. He is currently interning at Siemens in the Major projects and services division. He will continue as a full-time project engineer at Siemens upon graduation.

References

[1] *HC-SR04 Ultrasonic Range Sensor on the Raspberry Pi.* July 3rd 2014: https://www.modmypi.com/blog/hc-sr04-ultrasonic-range-sensor-on-the-raspberry-pi

[2] MIPI Alliance. *MIPI Alliance Working Groups Overview* http://mipi.org/

[3] Bill Earl, *LSM303 Accelerometer + Compass Breakout* Adafruit: https://cdn-learn.adafruit.com/downloads/pdf/lsm303-accelerometer-slash-compass-breakout.pdf

[4] *HS-322HD Standard Heavy Duty Servo* Hitec RCD USA, Inc. http://hitecrcd.com/products/servos/sport-servos/analog-sport-servos/hs-322hd-standard-heavy-duty-servo/product

[5] Total Phase, *Spi Background* http://www.totalphase.com/support/articles/200349236-SPI-Background