Autonomous TankBot

Stephen Greene
Electrical Engineering
University of Central Florida
Orlando, United States of America
steve.greene991@knights.ucf.edu

Anthony Speake
Electrical Engineering
University of Central Florida
Orlando, United States of America
anthonyspeake@knights.ucf.edu

Conan Wilson
Electrical Engineering
University of Central Florida
Orlando, United States of America
conanbwilson@knights.ucf.edu

Emmanuel Sotomayor
Electrical Engineering
University of Central Florida
Orlando, United States of America
esotomayor@knights.ucf.edu

***Abstract*—TankBot is a weaponized, autonomous vehicle that serves the purpose of defending human life from dangerous confrontation. The authors propose to do this by creating a system that can be used to replace emergency responders or military personnel for critical response situations. These situations include, but are not limited to: seek and destroy missions, active shooter handling, perimeter defense, and target tailing. The system will have the ability to identify potential foes and differentiate them from known allies, and in this sense won't require a manual operator throughout system use. However, the authors have found that most autonomous weapon systems currently available do have manual overrides, so TankBot can be manually (non-autonomously) controlled. The group sought to create a model device that would represent future innovations in autonomous defense systems, and thus recognizes many of the downfalls of the system. The broad majority of these downfalls are influenced by the lack of ample time for research, design, and testing, as well as monetary constraints upon the group members. That being said, this system represents a cost-effective model for others to build upon and improve as better technological alternatives emerge on the market.**

Keywords—Autonomous Vehicle, Autonomous Turret, Autonomous Defense Robot

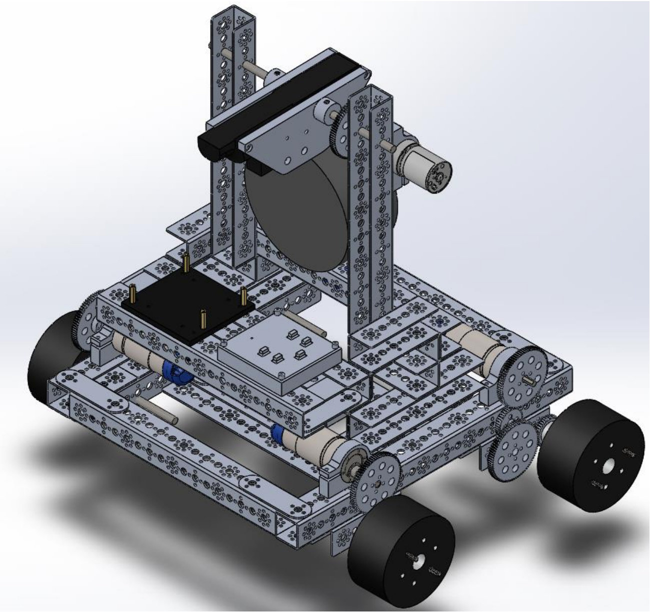


Figure 1 TankBot 3D Rendering

I. INTRODUCTION

The objective of this project is to design and build a fully autonomous mobile robot (Figure 1) that can correctly identify and neutralize targets. The function of the project is, then, to increase security in required areas while limiting human intervention so that the relative safety of friendly personnel can be maintained. Project features stem from comparison to autonomous vehicles that already perform this task (drones, automated turrets, etc.) as well as features we believe are missing from these products. The bot should remain lightweight, low cost, relatively low power, accurate, and implementable within the time frame provided.

The key advantages of TankBot's use include: reduced risk to human lives, lack of emotional inhibition, elimination of human inaccuracies, and the robot's ability to make thorough, well-thought-out decisions quicker than a human. As an elaboration of the objectives, TankBot should be able to: identify and differentiate a human target from the surrounding environment, determine the distance to a target, move effectively around several types of terrain, aim and shoot an airsoft gun to respond to hostile infiltration of a secured area, eliminate targets, and follow a target without shooting.

The engineering specifications involve: an accurate distance range measurement of at least 30 feet, a target hit rate of at least 25%, an effective range of at least 30 feet, a runtime of at least 30 minutes, real-time position tracking of the turret, and omnidirectional movement of the entire robot. In addition, the robot should be: remote controllable, able to stream video to a remote device, able to switch between autonomous and manual modes, able to display captured image from camera to an external display, and able to identify and differentiate targets. The overall system design can be viewed in Figure 2 and Figure 3.

One of the many motivations for this project was the outcome of the Pulse nightclub shooting on 12 June 2016. Pérez-Peña, Robles, and Lichtblau of the New York Times writes:

> "Many questions persist about those three hours at the blood-drenched Pulse nightclub, and about how law enforcement handled the crisis on June 12. Orlando police officials have been peppered with queries from the public, survivors and the news media about whether they should have confronted the gunman sooner and whether any of the victims were shot by the police." [1]

The level of skepticism and criticism over the decisions made that night highlight one of the key motivations for this project: TankBot will be able to make swift decisions about who the threat is and neutralize them without harming civilians. To that degree, it isn't odd for people to question whether a team of autonomous, turret-wielding robots could have better handled the situation than live people who are subject to their own doubts, uncertainties, and indecision. Of course, in the specific light of the Pulse nightclub shooting, officers' lives would not be at risk and there wouldn't be any skepticism about who was shot and what they were shot by.

The authors would also like the readers to recognize that this project was an opportunity for the members to dabble in recent emergent technologies, such as LIDAR, electronic airsoft integration, motor control, computer vision, and power management with integrated devices. Two of the major technologies in microcontrollers for TankBot stem from the creation and continuous improvement of the Raspberry Pi and Texas Instruments' ARM based TIVA microcontroller. The introduction of full rotation LIDAR units, OpenCV (as an open-source computer vision library), and fully electronic airsoft guns serve as important keystone technologies that allow for the creation of a system such as TankBot.

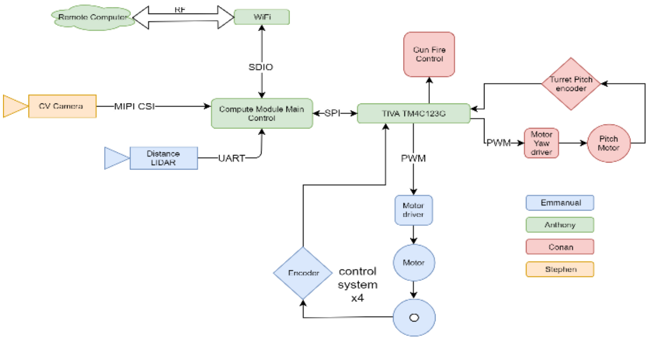


Figure 2 Overall Hardware Control

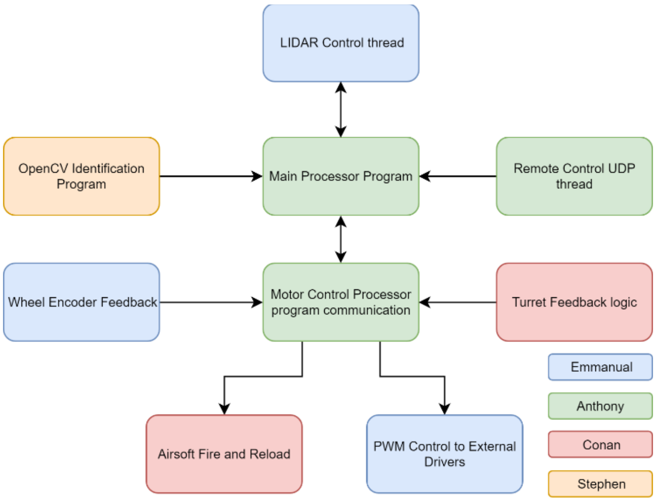


Figure 3 Overall Software Control

II. HARDWARE

The master control system works as the hub for all communication and decisions across the various subsystems and block components of the system. It is implemented with a Broadcom BCM2837 which utilizes an ARM Cortex-A53 microarchitecture. The computer vision will also be directly implemented on this microcontroller, as well as the calculations related to the LIDAR.

The turret pitch motion control system can be seen on the right side of Figure 2. The encoder provides feedback to control exactly where the turret is aiming, to accurately aim the turret at the correct target. The motors will the adjust as needed by the master control guided by the encoder feedback. However, the TIVA controller will be issuing the direct commands and computations related to turret pitch.

In the bottom of figure 2, the wheel controls can be seen. Mecanum wheels will be implemented, which require independent motors for each wheel. These motors will all be controlled by a motor control subsystem. The motor control subsystem is operated with TI'S TIVA TM4C123GH6PM microcontroller. Direct communication between the BCM2837 and TIVA microcontrollers is vital to the success of the robot. Thus, communication will take place between the wheels and the motor control subsystem, then between the motor control subsystem and the master control system. The motor control subsystem acts as a buffer between the master control system and the wheels.

At the top of figure 2, the wireless link system for communicating with a computer can be seen. The purpose of this set of block components is to allow the robot to interface with humans to some degree. Users can see through the eyes of the bot, so to say, which can allow users to possibly understand the motivations behind the decisions of the robot, as well as the process of the decisions. It also allows for wireless control of the robot.

Also seen within figure 2 is the CV, LiDAR, and gun fire control subsystems can be seen. The CV will operate through a camera which is interpreted and controlled by the master control system for universal control of the robot in its decision-making processes. The distance LiDAR will allow the entire system to know when a target is in range for shooting and also allow for some basic object avoidance. The gun fire control system will work as a buffer between the master control and the gun, communicating with the master control system on when to fire and when to idle. The reloading hardware is included within the gun fire control since its circuitry is so similar.

A. *Power Subsystem*

The power supply subsystem is the circuits which will turn the raw battery input into usable power for the entire system. It is arguably one of the most important as a bad power system will not allow the robot to function. For both the 5V rail and 3.3V/1.8V dual rail power distribution systems, switching regulators will be used instead of linear regulators to maximize efficiency. An LTC2953 (Figure 4) is used to form the battery protection circuit, which will ensure that the implemented lithium polymer batteries don't drop below 9.1V. Voltages below this will cause permanent damage to the battery's recharge life. A MIC261201 (Figure 5) rail regulator is used to supply a 5 V rail to the main control of the robot as well as 3.3V for the motor driver processor. Since the BCM microcontroller consumes around 5W at full load, the 12 A maximum of this chip should allow for proper operation of the 5 V rail. To supply 3.3 V and 1.8 V power rails on the main controller board, the PAM2306 (Figure 6) is used. The 3.3 V will be used for more of the I/O, while the 1.8 V is required (but not used) for logic. The PAM2306 will be feeding the BCM microcontroller. Since this chip is used commercially for all

Raspberry Pi boards, it should be sufficient for the BCM microcontroller.

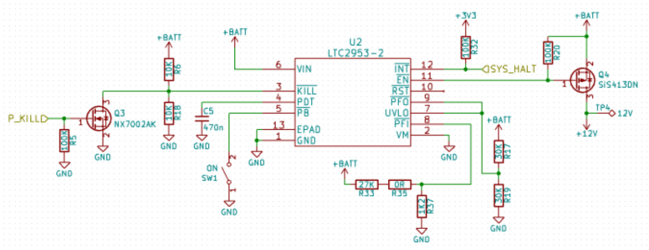


Figure 4 Battery Protection Circuit

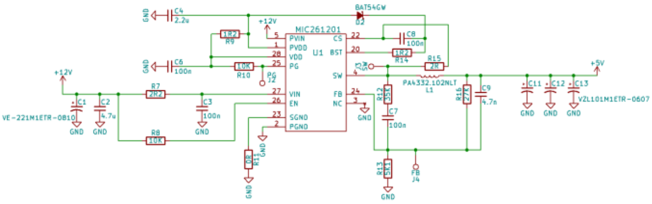


Figure 5 MIC261201 Circuit

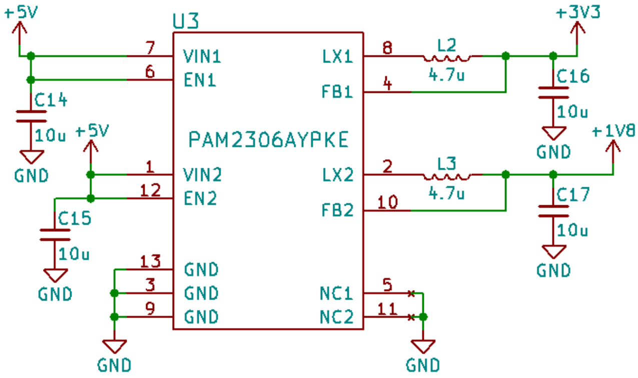


Figure 6 PAM2306 Circuit

B. Communication Subsystem

SDIO, CSI, and UART protocols will be used for the Wi-Fi, camera, and LIDAR respectively. SPI will be used to communicate between the BCM microcontroller and the TIVA microcontroller. For the high speed SDIO and CSI transmission lines, special care has been taken to reduce the delay between individual lines and impedance matched of transmission lines on-board the main controller.

C. Thermal Design

The Compute Module features a fully integrated SoC (System on a Chip) that contains both a processor and GPU in a small 14 mm^2 package. Since the TankBot will be doing heavy video encoding for video streaming to the remote-control program, the CPU usage is at 100% most of the time. As such, the SoC becomes very hot to the point that the chip will throttle itself, reducing core clock to reduce temperature. To prevent this, a heatsink is needed to efficiently transfer this heat from the chip to the ambient environment. Unfortunately, we were unable to find the specific $R_{\theta JC}$ junction-to-case thermal resistance. We tried both contacting the manufacturer SoC Broadcom or the device manufacturer but were unable to ascertain any information. Thus, a custom Heat-Sink will still be used based of an analysis of current commercial heatsinks for this SoC (Figure 7) Unlike other commercially available heatsinks for this processor, this utilizes the in-built mounting holes to secure the heatsink instead of glue. Not only does this provide better thermal performance as glue is a poor heat conductor, it allows for the easy installation and removal of the heatsink with screws. Thermal paste will be used between the heatsink and SoC to further improve thermal performance.

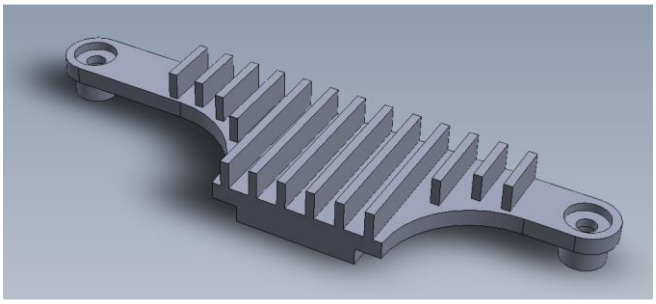


Figure 7 Heatsink for BCM Microcontroller

For the MIC261201, an exposed thermal pad directly under the junction to the PCB ground plane allows the small amount of heat to be dissipated properly. Thermal vias are used to properly dissipate the heat through all the ground plane layers to satisfy the PAM2306.

For the motor driver board, thermal pads will be used underneath this chip to move the heat from the top plane to the bottom plane (Figure 9). Since the driver can have up to 6A of drive current a heatsink (Figure 10) will be attached to the bottom of these exposed pads to conduct the heat from the PCB into the metal frame.

D. Motor Subsystem

The DRV8412 Motor Driver (Figure 8) has been used for all positional motors within TankBot. All positional motors within TankBot (locomotive and turret) will be controlled by these motor driver chips, led by the TIVA, for proper operation. The TIVA itself will be fed instructions on exactly how the motors should function by the BCM, in order to perform proper object avoidance and 'follow target' features.

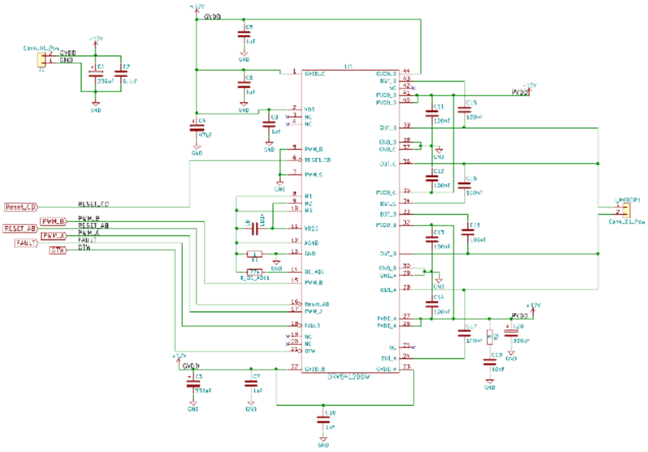


Figure 8 DRV8412 Driver Circuit

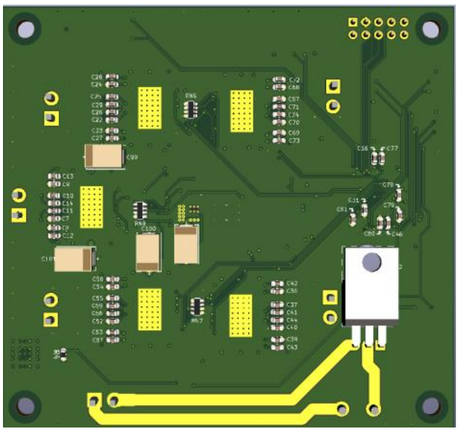


Figure 9 PCB Layout for Motor Driver Heat Dissipation

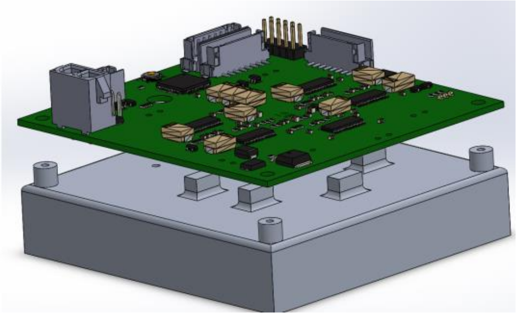


Figure 10 PCB With Heatsink for Motor Driver Heat Dissipation

E. *Gun Fire and Reloading Subsystem*

Unlike most of the motors inside of TankBot that adjust positional components of the robot, the turret and magazine don’t require pulse-width modulation. As a result, a simple transistor-controlled relay and fly-back diode are all that is needed (Figure 11). This will allow the turret or magazine to be digitally switched ‘on’ or ‘off’. To that end, no ‘strength’ of the motor needs to be adjusted by pulse-width modulation. The fly-back diode is included because the relay coils are inductive, thus when the switch is turned off, a back EMF causes a reverse voltage to be induced across the terminals. This voltage as large spike occurs once the relay is turned ‘off’. This spike is suppressed by the diode which becomes forward biased, allowing for safe operation of the relay controlling the motor. The GPIO pin that controls the base of the BJT is on the TIVA microcontroller. However, the TIVA will be receiving the fire command from the BCM, guided by the CV.

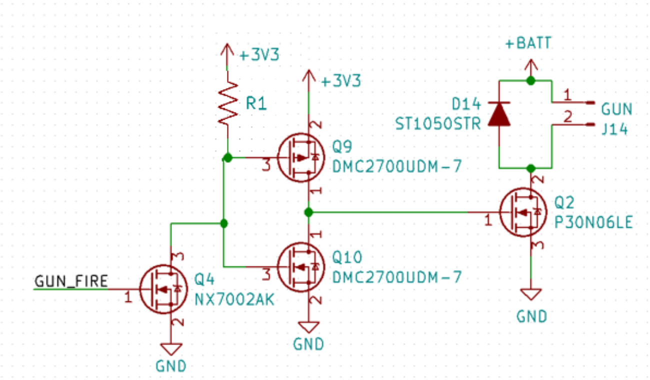


Figure 11 Gun/Reloading Switching

The airsoft gun chosen for the project was the M4 Airsoft AEG with Metal Gearbox (Figure 12). The main reasons the authors picked this gun over the numerous other options are the high durability of the interior components (being made of metal rather than plastic) and the inclusion of a high torque motor. The other guns were viable options, but their plastic, non-durable builds disappointed the group. Ultimately, a more expensive, better built gun will reduce the number of times it will be replaced. The weight was one of the highest of the options, which is a huge downside, but was aptly worked around. The capacity isn’t the highest, but was extended with an electrically controlled 1500 BB drum. The muzzle velocity is the highest of the relevant options, which harkens on the potential accuracy of this gun. The gun was received at a reduced price (around 50%) from the boneyard on Evike (where defective airsoft guns are often sold). After an investigation, it was found that the problem with the gun was purely aesthetic. This isn’t an issue since most of the gun was stripped down for the project.

Figure 12 M4 Airsoft AEG with *Metal Gearbox*

F. *Feedback Encoder Subsystem*

Encoders will be used throughout the system to monitor the present angle of the turret or position of TankBot. They will also be used to monitor the velocity of the motors. For each of the Mecanum wheels, the velocity and direction of the motors is of the utmost importance. This importance stems from the fact that TankBot will need to move around in a planar sense (sometimes taking adjustments in its skyward component, if traversing a sloped plane). Additionally, the Mecanum wheels allow TankBot to make turns without rotating around an axis system, but this is incredibly contingent on the synchronization of the wheel’s velocities and rotation directions.

The turret’s pitch is much less sensitive to encoder measurements, in the sense that there must be a maximum angle up, a minimum angle down, and the current angle of the turret must be monitored for aiming it properly. However, to that degree, the TIVA must only know the current direction of the turret and what angle it’s at. It doesn’t necessarily need to know the velocity of the turret.

Precise control of the encoder’s readings is important for both the wheels and the turret. The encoder for the wheels and turret will monitored with edge detection GPIO interrupts and a timer will give a velocity value for the wheels.

G. *Concluding Remarks Concerning Hardware*

Overall, the hardware of the system is designed such that the BCM microcontroller can control all imperative functions of TankBot and act as a high-level interface for the remote-control software. The TIVA microprocessor is used as a low-level management engine to cover the operation of the motors and encoders in real time. It also provides a buffer between the main processor and most of the peripherals of the system. The main functions of TankBot can be conceptualized as data collection from encoders, CV post-processing, and LIDAR working together on the BCM to form a decision about where to go, how to aim, and when to shoot. This decision is then sent to the TIVA to be processed as commands for the motors, which then move the robot, aim the turret, and fire the gun. This process forms the autonomous nature of TankBot. While under manual (non-autonomous) control, the data collection and decision-making processes are reduced or eliminated and TankBot becomes a hub for human action. Either way, the removal of direct human intervention from dangerous situations is ensured.

III. SOFTWARE

The software design for this robot will encompass all control and logic needed to achieve the goals that were generated at the onset of TankBot's creation. The master control for TankBot will be coming from the compute module which allows for the use of threads in Linux to achieve concurrent control of various subsystems of the robot simultaneously. The software is divided into two main categories: the main processor software and auxiliary control software. The idea behind this division is to separate the real-time, low-level management of various hardware interfaces with the high-level manager of the Compute module.

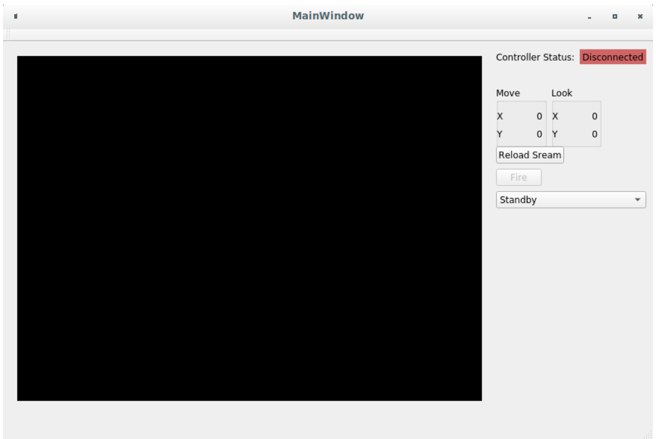
The main program will make use of multithreading to allow for simultaneous management of connected peripherals and receiving of commands. The software control diagram identifies the major components of the system at large: remote command, wheel control, turret control, and CV. The software control diagram allows users to see exactly what major components of the system exist within the system. The remote command allows for wireless communication with the bot to control its states. The wheel control allows for planar motion of the bot and control of the planar motion. The turret control allows for aiming and firing the turret at targets. The CV control allows for acquiring and differentiating targets.

A. *Main Processor Software*

The main control software will run on the BCM2837 on top of the of the Linux kernel The use of threads allows for processes to be executed asynchronously and the multi-core architecture allows for simultaneous execution.

The main control software will need to read information from the LiDAR, send commands to the movement controller, and process data from the remote CV. While this could be done in a single process, it would be extremely slow and inefficient. A better solution is to segment each task into an asynchronous thread which handles the interfacing with the various subsections of the robot and only reports back pertinent information to the main process. The Linux implementation of threading is called pthread short for POSIX Thread, which allows a program to control multiple different flows of work in the same overlap of time. By moving the LiDAR probe, and networking command interface into separate threads, they can run in parallel and the main can manage the data flow from each thread executing decisions based on received information.

To communicate between the main program on the BCM2837 and the remote-control application, web sockets will be used. All the control software for the TankBot is written in pure C, while the remote-control interface is written in C/C++. The benefit of using web sockets for communication between these two programs is that it makes the source language irrelevant and only the transmitted data structures matter.



Controller Status: Disconnected

Move
X 0
Y 0

Look
X 0
Y 0

Standby

Figure 13 Remote-Control Interface

The remote-control program is what will send commands from a host computer to the TankBot. Additionally, the remote command program will show the view from the TankBot's camera to allow for control without directly viewing the bot. During initial development, the team had the choice of two GUI based application development environments: Qt and .NET. Qt is a GUI framework written in C++ and licensed under GPL, while .NET is made by Microsoft with C# and has native GUI frameworks built-in. Qt was chosen as it provides fast native code execution unlike C# which runs in a VM and cross platform support since the control application will run on a Linux host. The preliminary remote-control interface can be seen in figure 11. The manual remote-control will be done via an Xbox 360 controller.

Although the Wi-Fi module connects through a standardized MMC interface, it is impossible to know how to communicate with every single possible type of hardware both past, present, and future. Therefore, drivers are needed which give a standard interface for the kernel to create the abstracted interfaces that higher-level software can expect and use. The driver is a kernel extension that translates the specific and sometimes proprietary controls of the device to the kernel. For the WILC3000 wireless module, the manufacturer's driver was used as a base and then modified to properly support the application case. The driver initializes the device, waking it from sleep, then registers the device as a standard interface for higher level code to access. Any time a program accesses the WLAN interface, the driver will communicate with the device over the standard SDIO protocol.

B. *Computer Vision*

OpenCV was the computer vision library chosen for TankBot. OpenCV will run on the control PC and relay information via the remote-control interface to the main control software on the BCM microcontroller. OpenCV's libraries include the code needed to execute many of the most important aspects of this project including target detection, target identification, motion detection, and many others.

The motion detection systems work by observing the differences between any two frames in sequence and ignoring anything that is the same. After taking the difference between the two frames the output will be shown as a screen of white pixels indicating that those pixels are not the same and thus something has moved in that location. By doing this process repeatedly TankBot can track the motion and the shape of whatever is moving.

Target detection is a much more complicated process called the Histogram of Oriented Gradients and Object Detection. This process consists of showing the software, which is provided by OpenCV, several positive samples, showing several negative samples, where the number of negative sample images is much greater than the number of positive samples. A set of descriptors will be applied to each of the images such that each image is codified: this will be important to the next step. Apply a support vector machine to each of the images descriptors, this machine will draw a "line" between what is a positive image and a negative image. After this step, application of Hard Negative Mining will ensue, which is the process of letting the program attempt to find which parts of an image make that image a positive. This occurs by correcting the false positives found during this process. Thus, the classifier is taught how to better differentiate between those negative and positive images. The process can stop here, but the more implementation of hard negative mining, the better the classifier will be. Thus, the percentage of false positives and false negatives will continue to decrease. These pre-trained classifiers are available online through the TensorFlow API. One of these pre-trained libraries has been edited to improve performance for our task, specifically, we are only interested in detecting human beings.

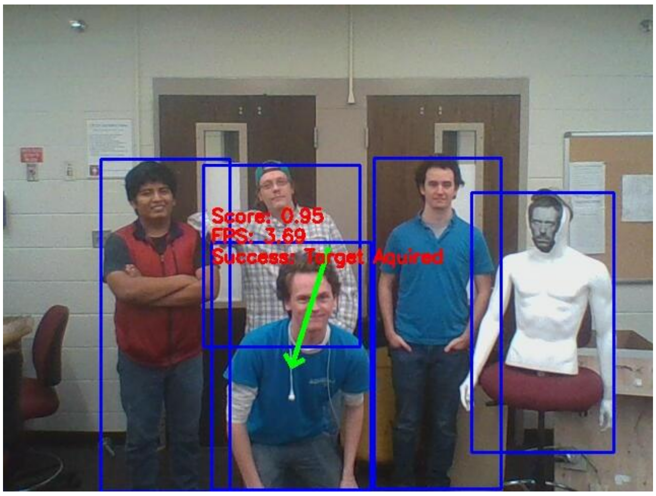


Figure 14 Target Identification Example

The process for identifying targets is very similar to the process for detecting a target in the cameras vision, it just has a stricter view on what is a positive hit and what is not. Rather than any human-shaped thing being flagged as positive, only human-shaped things that also are holding a gun-shaped thing are flagged. In addition to this method, the machine can pay attention to how close the human shaped thing is to the robot, or even how fast the human shaped thing is moving in any direction to further classify targets and differentiate friends from foes. Figure 14 shows an example of how the OpenCV would function on the TankBot. In this example, no filtering is performed, and any humanoid target is identified. Work must be done to not only classify targets but also to increase the detection rate and precision. This example outputs whether or not the system has found a valid target, the FPS of the camera, and the score of the focus of the image, which is an indicator of how sure the CV is that the found target is a person.

C. *LIDAR: Object Avoidance Radar and Follow Logic*

While the computer vision can be used to distinguish people from other objects, The LIDAR can distinguish distances from objects in its surrounding areas. When using a direct LIDAR the return from the device is a point in front of the machine. The return of the reading also has the intensity of the beam. The intensity of the beam is an estimate of how sure an object is in front of the LIDAR.

For objects closer to the LIDAR, the point have a greater point intensity, as oppose to objects farther away that have a lesser intensity. One thing to consider is that objects not in its direct vicinity of the LIDAR will not be picked up by the scanner. This could lead to miss-readings where TankBot runs into things that are just off angle of the LIDAR sensor. To compensate this flaw, the camera of the

D. *Gun Firing and Turret Pitch Logic*

The gun firing logic is quite simple. After the CV has acquired a target and the pitch of the turret has been set, the TIVA needs only to drive the port of the fire trigger high. When shooting is the cease, it needs only to clear the port. The BCM will communicate with the TIVA through SPI on when to fire and when to hold fire.

After a target has been acquired through the CV, the BCM will need to send the required angular position of the turret to the TIVA. The communication will be conducted through SPI, and the numeric code sent to the TIVA will need to be interpreted by the TIVA. The TIVA will then adjust the turret pitch as needed, based on the present angle (tracked by interrupts generated by an encoder). Afterwards, the MCU begins launching the gun firing logic described above. The pitch is adjusted through PWM signals sent to the motors that control the pitch of the gun. Thus, all control of the turret motors is performed by the TIVA, while instructions about what state it should be in is sent from the BCM. The reloading of the gun utilizes the same code as the firing of the gun.

E. *Locomotive Logic*

The locomotive control of the robot will operate in 4 primary modes:

- Standby
- Patrol
- Follow Target
- Attack

In the standby mode, the robot will simply halt in place. The locomotive motors will not be active, so the robot will not drive around. As such, the robot will simply be holding its ground in the area. The robot will also not shoot targets while in this mode. Essentially, it's a sleep mode for the robot.

In the patrol mode, the robot will trace a perimeter using the CV and LiDAR to guide it. The BCM will control the patrolling actions to activate and deactivate the locomotive motors as necessary to allow the robot to seem as if it is patrolling an area. If a target is acquired by the CV, after rating distance with the LiDAR, the robot will aim the turret and shoot the target.

In the follow target mode, the robot will have acquired a target and will follow it without shooting unless the user specifies otherwise, or the target imposes an approaching threat to the robot. The BCM will control the following using the CV and LiDAR as a guide for deciding on how to drive the locomotive motors and whether to activate the firing system to shoot the target.

In the go to location mode, the robot will simply travel to the target location. The locomotive motors will be activated or deactivated, as necessary, by the BCM to allow for routing to the location. The robot will be guided by CV and LiDAR. As the robot approaches the location or arrives at the location, if an enemy is determined by the CV, the robot will fire if the enemy is within the robot's effective range.

These modes will be specified by the user through the remote command logic software subsystem across the wireless link. As such, the user will have direct control over the actions of the robot and whether it is acting autonomously or not.

The method to accomplish this is to receive one control byte, 4 bytes of movement data, and one byte of state data, from the master controller. These data hold the coordinates for the next position in relation to the robot. With this the robot will align to a straight path to it and follow the straight path. Should the path be obstructed by any immediate object, the reactive sensor will push the robot to another direction. As a result, the displacement would need to be considered. To reduce the amount of error, the robot will need to be able to decide the path with the least cost to its effort. Therefore, as an overlay to the robot moving, a vector field with all the vectors pointing to the desired location can be implemented to move the robot to its desired location. Figure 15 is one such implementation of a vector field.

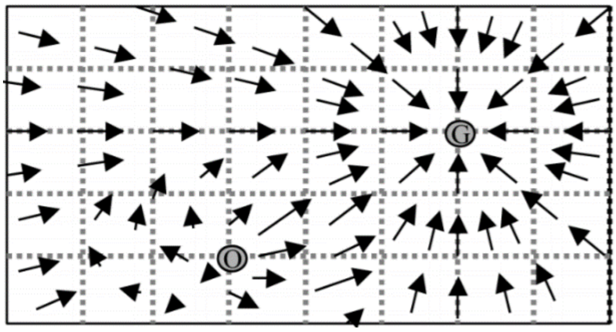


Figure 15 Vector Field of Robot (O) Guided to Goal (G)

One thing to not do is use absolute directional vectors and add noise to these fields. By not having noise inserted, should the robot enter a three-walled corner, then the robot may perpetually get stuck trying to recalculate the directional vectors. By adding noise, this adds randomness to the vector field and can help the robot eventually leave these chokes. The tools required for this would be the LiDAR scanner with odometry feedback coming from the encoders.

IV. CONCLUSION

Since the dawn of the microchip the technologies surrounding humans has become more and more integral to allowing mankind to experience the fullest of the human experience. Machines make human furniture, harvest human crops, and do some hundred-thousand other tasks that humans have deemed too menial or degrading for a human to have to do. However, all over the world, military, law enforcement, and even civilian lives are put at risk every hour of every day. From bank vaults to battlefields, humans trust only human beings to be the last line of defense when confrontation arises. Threats need to be assessed and dealt with in a timely manner.

New technology is being developed every day to take more and more of the responsibility out of the hands of human beings and into the more precise, unfeeling, unerring hands of machines. New-age electronic devices have all the capabilities to identify, target, and process images and objects at many times the rate of the human brain, all while being un-phased by the less than helpful feelings that plague humans, such as fear for one-self, indecisiveness, or any other emotion that prevents humans from acting when they are most needed.

The goal of this project aims to combine the emerging technological field of computer vision with the efficient motion and decision-making power of today's machines to eliminate the need for human combatants or first responders in the day to day operations and missions required to keep society functional and the enemies at bay. Some of these missions include but are not limited to: patrol and identification of targets, guarding a location from hostile forces, and seek and destroy missions in some of the most dangerous places in the world that humans are constantly risking their lives to protect. These tasks are eminently solvable by modern machines, with the processing power and agility of decision making allowed by the parallelism in the technology of today.

To meet the goal of being able to remove humans from these very dangerous situations, the authors have developed an autonomous wheel-based robot: TankBot, which will, in the future, be able to replace soldiers on the frontline, guards in high security areas, and even law enforcement officers when the need arises. To be able to do all of this, the robot needed to be designed for a couple of important objectives like being able to move quickly and independently of an operator over rough terrain. Also, TankBot needed to be able to identify not only the shape of a person, but also of a weapon that might distinguish a person as a foe. Additionally, it needed to be able to track a moving target and follow it with the option of compensating for that motion and then targeting and firing the on-board weapon at the target. TankBot had to accomplish all these tasks, sometimes simultaneously, without any human intervention beyond the command of what mode it should be in.

Meeting these goals did not come easily to the authors and several challenges were faced. One of these challenges was that the original frame built by the group wasn't sturdy enough to meet the demands of the project. Furthermore, it restricted the available space for a larger magazine on the gun. As such, a new frame had to be sought and the group found a person willing to lend a suitable frame to the group for the duration of the project. Another important issue encountered by the group involves the BCM not being able to effectively run TensorFlow (a faculty of the computer vision) at an acceptable frame rate. As a result, the system wasn't able to effectively process images and make decisions. The solution to the problem was to have the camera stream the footage through an internet protocol, while processing the images externally. The movement vectors would then be transmitted back to the system in order for TankBot to act as needed. However, even this solution came with its fair share of problems given that the most appropriate laptop's graphics card and BIOS had many incompatibilities with drivers outfitted for use within a Linux environment. The lack of familiarity with this issue as well as the remaining time until the final deadline for the project motivated the group to use a less optimal setup to process the CV. The group originally opted to use MOLEX Micro-Lock 4-position headers because one of the authors had many of these connectors lying around and they seemed to work well for the LIDAR. However, it was very difficult to hook these up, so the group ended up having to switch to insulation displacement connectors instead.

The authors are generally satisfied with the outcome of the research and work invested into the project. Despite setbacks and design trades, a feasible initial prototype of the project's goals has been created and functions as proposed within the specification requirements. As mentioned before, there are many improvements that can still be made given more time and financial investment, and the group hopes that these will be researched and implemented in future updates to TankBot.

V. ACKNOWLEDGMENTS

We would like to acknowledge and thank Don Harper and Clarence Bess in the TI lab for letting us use the Tetrix robotics kit for two whole semesters, Mr. Staton for his experience in robotics and willingness to help whenever called upon, Dr. Chan for his insights into power electrons and general excellence in the field of engineering.

REFERENCES

[1] R. Pérez-Peña, F. Robles and E. Lichtblau, "Orlando Police Defend Actions as Clock Ticked in Massacre," New York Times, 20 June 2016. [Online]. Available: https://www.nytimes.com/2016/06/21/us/fbi-transcripts-orlando-shooting-omar-mateen.html. [Accessed 7 July 2018].

A. Authors and Affiliations

Stephen Greene is a senior at the University of Central Florida studying Electrical Engineering. He has worked in and is interested in continuing to work in the field of robotics.

Emmanuel Sotomayor is a senior at the University of Central Florida studying Electrical Engineering. He has worked in and is interested in continuing in the field of control systems and embedded systems.

Conan Wilson is a senior at the University of Central Florida studying Electrical Engineering. His interests involve filter design, quantum computing, and communication theory.

Anthony Speake is a senior at the University of Central Florida studying Electrical Engineering. He has worked in and is interested in embedded programming with Linux systems