Traffic Stop Watchdog

Jordi Niebla, Felipe Solanet, Michael Gendreau
Dept. of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida, 32816-2450

***Abstract* — due to recent events, it has been noted that having a law enforcement job can be quite stressful. Many police officers have encountered many scenarios where they have had to react quickly and be aware of their surroundings. Whether it is a traffic stop gone wrong or getting a call about a situation where not much information is presented, a lot can occur during an interaction between a police officer and the public. The "Traffic Stop Watchdog" project is a design proposal to fill a current gap in the information recorded during interactions between police officers and citizens during traffic stops or other exchanges within the vicinity of a police cruiser. With the implementation of AI, we can use image recognition to track and follow a police officer with 180-degree rotations. This will allow for high resolution and high frames per second recorded footage with the police officer always in frame. The system will also include a handheld wireless microphone device that the police officer will have with speech recognition technology. The handheld device will be responsible for enabling and disabling the system, and detecting keywords said by the officer.**

***Keywords* — AI, Image Detection, Speech Recognition, FPGA, Odroid, Stepper Motors**

I. INTRODUCTION

Law enforcement personnel work in a highly reactive environment that demands focus and attention to avoid life-threatening situations. The ability to rapidly respond to changes in law enforcement scenarios is imperative, and any lack of situational awareness and information capture in these critical moments inevitably leads to obfuscation of events in the ensuing judicial review. Recent events have shown that for the safety of both law enforcement personnel and the public, higher precision information is needed in this line of work.

The "Traffic Stop Watchdog" project is a design proposal to fill a current gap in the information recorded during interactions between police officers and citizens during traffic stops or other exchanges within the vicinity of a police cruiser. Current systems like bodymounted cameras and dashcams often leave much of the scene out of view or out of focus while lacking support for live data communication. This project aims to develop a vehicle-mounted information system that utilizes AI to track a moving target on camera while providing support for a wireless speech recognition device that can be carried by an officer.

Driving the overall requirements for this system are the factors associated with its use. To be vehicle mounted, the design must be lightweight and compact with a strong enough chassis to withstand vibration and acceleration. The camera electronics must be powered off a standard car battery, consuming as little current as possible when the vehicle is off so as to not drain it. The speech recognition device should be a low-power device embedded in a small form factor design so as to avoid weighing down the user. Wireless communication between the devices in the system should maintain a minimum range of at least 500 ft, preferably greater.

By utilizing machine vision algorithms, the system should be capable of resolving more important information in a scene than current cameras. The onboard camera will track the law enforcement officer so as to record only the information vital to the scenario. 180 degrees of rotation and high placement on the vehicle will allow tracking in all directions. To maintain track of a fast-moving target, the system will need high responsiveness and an accurate control loop keeping the camera centered. This will necessitate a blend of precision electromechanical design as well as robust software development.

II. HARDWARE OVERVIEW/SELECTION

In the Hardware Overview/Selection section, various hardware components chosen for our design will be discussed. These components are the main building blocks of our design, and in-depth decision making was needed to pick out these components.

A. Single Board Computer

In charge of running the main logic that controls the entire system, the single-board computer needed to be capable of communicating to multiple devices simultaneously and processing/recording frames at a respectable rate. The operating system requirement set forth by the use of the Flir Firefly DL limited the choice of CPUs that could be used. Offloading the actual object detection work to the camera allowed for the single-board computer to be more dedicated to recording and communicating with the devices. The Odroid XU4 was chosen to take on this responsibility. Its 5V DC power requirement could easily be supplied from the 12V car battery. It comes with a small fan over the processor to help cool the chip. Running Ubuntu 16.04.3 Mate, the scheduling and file IO was already set by the OS and could be utilized in the overall logic of the system.

With two USB 3.0 ports and one USB 2.0 port, the Odroid met the requirements for communicating with multiple devices. Unfortunately, the enclosure built for the Odroid 15 blocked off the USB 2.0 port, so an adapter was created to allow two USB connections in one of the ports that will be discussed in a later section. Equipped with a quad-core ARM Cortex-A15 2GHz and a quad-core ARM Cortex-A7

1.3GHz CPU, the Odroid has the speed and multicore architecture required for fast response times and multi-device communications. A 64GB micro-SD card was used to hold the operating system and store a sufficient amount of footage. A convenient HDMI port and Ethernet port on the Odroid was used for debugging and setup purposes.

B. Field-Programmable Gate Array (FPGA)

Embedded logic in our design takes the form of microcontrollers and FPGAs. The primary communication bus, using UART, needs multi-slave capacity, which is implemented via a small FPGA. The camera assembly utilizes an FPGA as the motor controller, converting angular inputs over RS232 into driver signals for the motors. The choice for a Xilinx product was made for the camera motor logic module. Their development environment and debug/simulation interface is free for hobbyist use, and their products are equipped with more than enough logic elements for the hardware design needed for motor control. Additionally, if processing/filtering is needed on the angular error inputs received from the SBC, the Xilinx chips have DSP slices that boost digital signal processing performance.

The factors involved with choosing the specific Xilinx FPGA chip for our design included cost, package type, and peripherals. FPGAs can be very expensive devices, so the lower-end Spartan 7 lineup was our first choice. The 7 Series Spartan FPGA chosen is the XC7S6-1FTGB196C, which includes 100 I/O pins and 6000 logic elements in a 1.0 mm pitch Ball Grid Array (BGA) package. Routing a BGA package IC on a PCB is typically quite a challenge, but this is mitigated by the use of a 4-layer board and the minimum feature size for vias and traces provided by JLCPCB, our PCB vendor. To store configuration data, a Xilinx-compatible 32kb flash SPI chip is also needed, and an external 12 MHz oscillator provides the system clock for the Spartan 7.

For the UART bus manager FPGA, a Lattice Semiconductor device was chosen. Lattice specializes in small-scale FPGAs for single-purpose embedded logic. The design calls for a simple hardware configuration for buffering, receiving, and sending UART messages from different devices depending on message priority and address. The specific device chosen was a Lattice MachXO2-1200, which packs 1280 logic elements in a 32-pin quad-flatpack no-leads (QFNS) package.The MachXO2 has an onboard clock and nonvolatile configuration memory, which shrinks its overall footprint on the space-critical PCB layout.

C. Microcontroller (Handheld-Device)

On the microcontroller end, our requirements are for a low-power, multi-peripheral MCU that controls various functions of a handheld microphone device. This will include a display, a microphone module, and power monitoring.

Two of the main reasons the MSP430FR6989 was chosen is because of the cost-effectiveness and low power consumption that the MSP430FR6989 has. It is simple to use, and throughout the UCF curriculum it is a MCU that was popular for use of embedded programming. It runs off a 16-bit RISC architecture with a clock of up to 16- MHz. The MSP430FR6989 uses 128KB of FRAM, which is a non-volatile memory that uses ferroelectric thin film capacitors to store data. This allows an increase in performance at lowered energy budgets.

The microphone device will communicate to an XBee module via UART, which then will wirelessly communicate to the Odroid, and we believe that the UART peripherals that are offered from this MCU are great for the use of our project. The eUSCI (enhanced Universal Serial Communications Interface) module on the MSP can include the two channels, (channel A & B), which channel A supports UART and SPI communication while channel B supports I2C and SPI. I2C will be used to communicate with the LCD on the handheld device.

D. Electromechanical

The electromechanical components of this design include motors for movement in two axes as well as the gearing and associated driver circuitry in the camera assembly. The motor technology choice is at the heart of this section of the design - the solution needed must be compact, precisely actuated, and provide enough torque without drawing too much power. To achieve this, our design weighed the options of brushed DC gear motors and stepper motors.

Stepper motors were the most cost-effective solution, winning over the gear motor solution due to their precision control and compact size. Because stepper motors can be controlled “open-loop” without additional sensors, their size and cost are kept small. Additionally, the rise of mass-market 3D printer technology has pushed the price of stepper motors lower. The only downsides of stepper motors are that they require power to “hold” a position and the control logic to commutate them is more complex. To drive a multi-phase motor, a commutation sequence is required on the inputs to advance the shaft angle. Luckily, integrated circuits exist which perform the commutation from internal look-up tables, directly powering the motors and reducing the complexity of the required controller. For this purpose, the Texas Instruments DRV8846 motor drivers were chosen.

E. Camera

A big revolution in the camera search came from a Flir Systems' Firefly DL camera. Producing 1080p images at 60 FPS over USB 3.0 and a compact, light design, this camera definitely met our projects requirement. With greater specifications than the GigE cameras came a larger price however, costing about 1.5x more. What we believe justified the significant increase in price and ultimately led to the selection of the Flir Firefly DL was not in fact the impressive specifications, but the camera's built-in image processing.

The camera is the first in the industry to implement the Intel Movidius Chip, a vision processing unit (VPU) built to process images through a deep-learning model at competitive speeds. This chip would allow the object detection to be done on the camera and significantly increase camera response rates while slightly lowering the programming effort. A lens was not included with the camera, and so a bundle containing a 6mm, 8mm, and 12mm lens was purchased. The different focal lengths would allow for more flexibility and customization when transitioning from a test environment (a small room) to the production environment (outside).

F. Mechanical Design

To support the fast tilt and pan rates needed for an autonomous camera, a rugged and stable two-axis platform is needed. As a foundation for this design, our team purchased a cheap pan and tilt mount for closed-circuit surveillance cameras from a surplus store. The mount, equipped with geared AC motors and held together with plastic screws, was initially unsuitable for our purpose. It lacked rigidity in both axes, and the assembly was flimsy with lots of play in the gear subassemblies. It served as a great starting location, but much of the original hardware was replaced or redesigned for our usage. This was known ahead of time since the original mount was designed for manual control in 90 degrees of movement for both axes.

To accommodate the face profile of the stepper motors to the mounting locations for the old motors, 3D printed faceplates were designed and printed for the steppers that precisely aligned the motors to where their shafts needed to be based on grooves in the plastic housing. Instead of the self-tapping plastic screws that held the original motors, new stainless-steel fasteners were installed, in 4 locations instead of 2, with metal-on-metal thread engagement to securely retain the motors. Seen below in Figure 1 is the modified camera mount internals, that can include all the adjustments needed for 180-degree pan rotation, specification for our project.

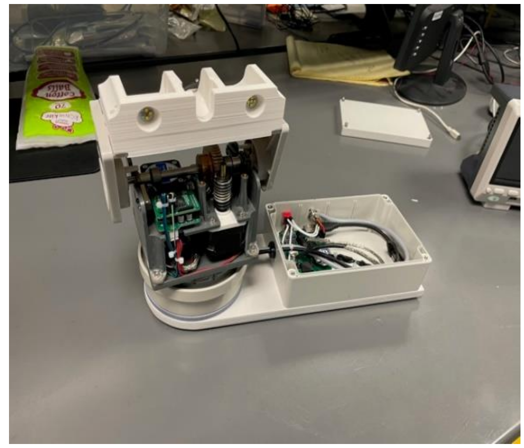


Figure 1: New Camera Mount Internals

III. HARDWARE DESIGN

In this section, hardware design will be discussed. This can include schematics and PCB layouts which were designed for use of the Traffic Stop Watchdog system. All PCBs were ordered and manufactured via JLCPCB. Below in Figure 2 shows the overall hardware block diagram for our project. It is composed of two main units, the microphone unit, and the camera unit.

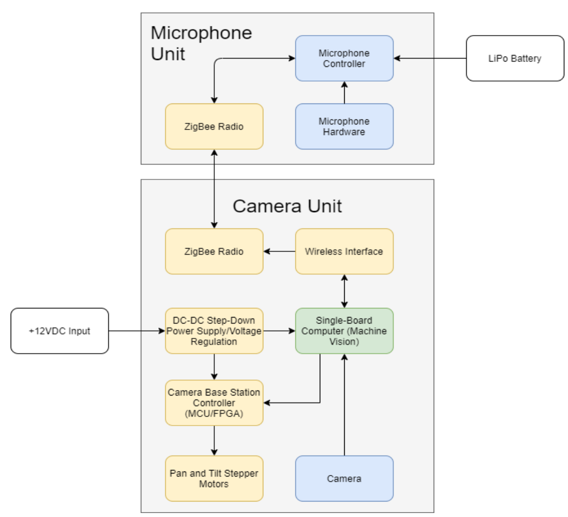


Figure 2: Hardware Block Diagram

A. Handheld Microphone Unit

For the design of the embedded microphone device, the speech recognition module needs somewhere between 4.5-5.5V to operate on, while the MSP430FR6989 chip & the Xbee module need the VCC supply to be at about 3.3V. This means that a voltage regulator will be needed to ensure that the proper stepped down voltage from the LiPo battery will be distributed to the MSP chip and other components on the PCB. The major components on this PCB board can include the MSP430FR6989, an LCD screen along with some push buttons for the user interface, boost converter, voltage regulator, battery charger, and an Xbee Series 1 module that will wirelessly communicate with the Xbee module on the Odriod's communication board.

Figure 3 shows the schematic for the handheld microphone device, with all the components. As seen below, the MSP430FR6989 has many pins, but not all the pins will be used. On the top left of the figure, we can see the boost converter that will feed 5V to the speech recognition module. Below that we can see the voltage regulation that will reduce the 4.2V Li-Po battery input voltage to a 3.3V output which will be necessary for other components. Other components also on the schematic can include the LCD, Xbee Module, LiPo battery charger, JTAG pins, and three push buttons. The MSP430FR6989 has two communication channels for UART transmission, USCI_A1 and USCI_A0. The speech recognition module will be connected through Channel A0, and the Xbee device will be connected through Channel A1. The LCD communicates with the MSP via I2C, and therefore will be connected through the USCI_B1 channel.

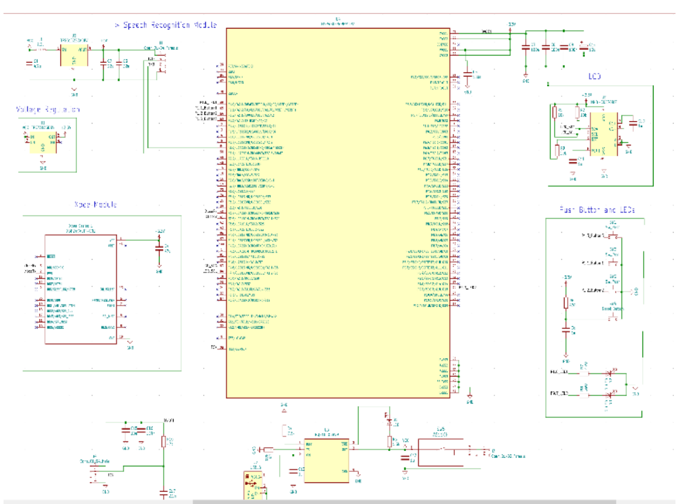


Figure 3: Handheld Microphone Device Schematic

Figure 4 shows the actual PCB and whole microphone unit. The dimensions are about 4 x 4 inches.

Figure 4: Handheld Microphone PCB

B. Camera Unit

First, we will begin with the power and communications board that is shown in Figure 5. A number of voltage supplies are needed for the various subsystems in the camera assembly. The primary supply is a 12V lead acid car battery found in most automobiles. 11.5V - 13V is fed to the power and communications board located within the SBC enclosure. This board is a 4-layer design, with the stackup specified as JLC7628. Primary input power is filtered by a pair of 100 uF electrolytic capacitors and protected by a 6A fuse. An inrush-current limiting NTC is in series with the main fuse to prevent high capacitor charging currents from damaging the vehicle power bus or blowing the fuse. Screw terminals are used to prevent vibration from loosening connectors.

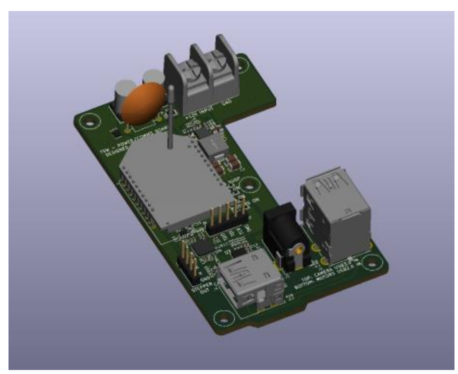


Figure 5: Power & Communications Board

From the filtered and protected input 12V supply line, a high-current capacity 5V supply rail is generated. This is achieved by a Texas Instruments TPS54824 switch-mode voltage regulator. Rated for 8A output current capacity, it provides the primary input power for the Odroid XU4 SBC and Firefly DL Camera. The output 5V supply along with the 12V supply are passed through three separate Texas

Instruments TPS2592 electronic fuses, one for each major subsystem power rail (5V camera, 5V SBC, and 12V motors). This protects each subsystem from voltage spikes on the primary 12V bus, prevents devices from drawing power if the vehicle battery is below the undervoltage discharge threshold (11.5V), and shuts down subsystems that draw more than their rated current consumption. Status of each subsystem enabled power input is verified by an onboard surface mount LED.

The power and communications board additionally serves as the interface between the high-level AI computation taking place on the Odroid XU4 and the low-level processing needed to drive the camera motors and provide user input from the wireless microphone device. Figure 6 is a diagram that explains the systems communication flow. The primary communication bus between these different subsystems is a modified UART protocol. The host, the Odroid XU4 SBC, needs to send and receive serial data to two subsystems using UART. The microphone device communicates using a pair of XBee Series 1 digital radios, which have onboard UART interfaces for the incoming and outgoing data. The motor module has a UART transceiver implemented on the driver FPGA that translates angular error commands into stepper movement.

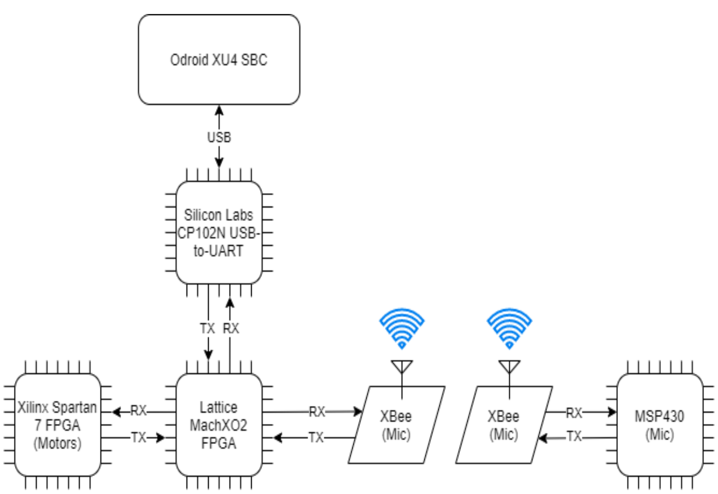


Figure 6: Systems Communication Diagram

Natively, the UART protocol does not specify an address scheme or provide for multiple slaves on one bus like I2C or SPI. Unfortunately, the need for UART was driven by specific hardware availability, so in order to make it work in this capacity, a small Lattice Semiconductors MachXO2 FPGA was implemented as a communication bus manager. The device allows two slaves to communicate with the SBC host without any messages being missed.

Next is the motor driver and controller boards. Because the motor driver and controller circuitry are very dependent on one another, the two boards are combined as a subassembly in a stacked configuration. The separate-board, stacked configuration is also beneficial because it simplifies the hardware development, reduces the footprint of the boards, isolates delicate logic and flash memory components from the more robust driver components, and isolates high voltages and switching power components from the low voltage digital controller FPGA. Figure 7 demonstrates the stacked configuration of the boards.

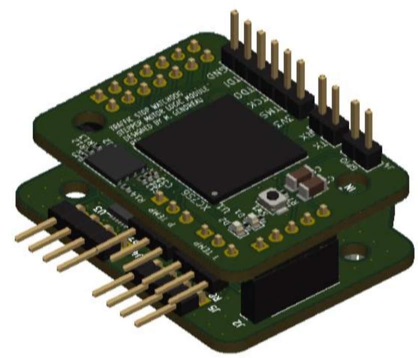


Figure 7: Driver-Controller Circuit Assembly

For the driver board, the selected motors are two-phase with 1.8-degree step resolution, which is sufficient for the angular resolution of the design given the gearing on both axes. However, for reduction of noise and vibration, the motors are micro-stepped using the commutation logic of the DRV8846 stepper driver ICs. These reside on the stepper driver and voltage regulation board, part of the main power and logic subassembly that includes the FPGA board which stacks on top in order to receive power and deliver motor inputs. Using the highest level of micro stepping capable of these chips, 1/32 of a full step, we are capable of achieving angular resolution of 0.05625 degrees in each axis before gearing.

In the controller board, the logic board for the stepper motors is a small (60x40mm) 4-layer board that mounts directly on top of the aforementioned driver board. This board receives UART commands from the Odroid XU4 SBC that translate to movement of the motors in proportion to the angle of movement commanded, and it also monitors various parameters to prevent damage to the assembly. The brains of the board reside in a Xilinx XC7S6 Spartan-7 FPGA with configuration data stored in a Macronix MX25L3233 nonvolatile flash memory IC.

The 4-layer stackup for this board is identical to the driver board’s stackup. 4 layers is more important for this board because the chosen FPGA IC uses a 196-pin ball grid array (BGA) package in a 15 mm x 15 mm footprint. Signal and power routing are therefore challenging to achieve with most board houses' 2-layer capabilities. The Xilinx FPGA also requires a great deal of decoupling capacitors for the various IO banks and internal voltage supplies, called for by the chip's datasheet. These are accommodated in the design by use of small 0402 SMD ceramic capacitors placed on the underside of the board, mounted as close as possible to the respective voltage supply pins on the FPGA. Decoupling capacitors are placed in order to "decouple" the attached device from potential noise on the voltage supply bus, as well as reduce the amount of noise put back into the bus by internal switching of the device.

IV. SOFTWARE DESIGN

Acting as the brains of the operation, the software was an integral component to make the system function as desired. The software itself has a lot of components, and so it needed to be organized and efficient to minimize error between its components and maximize ease of integration with the physical hardware. To accurately track the officer, an object detection model needed to be trained that could distinguish police officers from other people. The camera needed to be integrated into the recording process and the logic portion that controls where to move the camera based on the officer's location in the frame. To give the officer control of the system, a communication protocol from the microphone device needed to be established and followed.

A. Dataset Creation/Training

The ideal dataset would contain images both with and without police officers, some having multiple in the same image. Because the model would need to recognize officers in the middle action, it would have to be flexible enough on the orientation and pose of the officer in the image so that it could recognize officers that were not facing the camera directly and possibly moving. Lastly, multiple different environments/backgrounds would need to be present in the images to prevent the model from being too reliant on officers' contrast compared to the background.

Datasets for deep learning can be very large, containing thousands of images that represent the data. To achieve such a large number of images, videos were taken and split into frames. To meet the requirements for the dataset, a script was created that would mimic a typical traffic stop and capture the officers moving towards the stopped vehicle as well as standing still when talking to the driver. Two officer costumes were used to act as visual definitions of a police officer to the model. The following figure shows an example of a skit we made for our dataset.

Recorded Footage

After recording the footage for the dataset, the videos were split up into frames, producing a dataset with just under 2000 images. The total number of frames in the video far exceeded 2000, however most frames were skipped because of the immense similarity between adjacent frames. While this decreased the size of the dataset, it made the dataset more spread and allowed for each image in the process to be processed more in the same amount of time. With all the images extracted from the video, the next step was to label each image with the training metadata that would allow the algorithm to correctly identify police officers. A label making tool was used to generate all the labels for the dataset using the PASCAL VOC Format. Finally, the images and labels were compiled into an LMDB (Lightning Memory-Mapped Database), which was used as the dataset to train the model.

To help decrease the overall training time, the images were resized to a resolution of 1280 x 720. While this did decrease the detail contained in each image, it allowed for more iterations to be used, which provided a larger benefit than the small extra detail that was cut out. A new problem came up, however, that pointed towards a flaw in the dataset (as opposed to how the model was trained). The model pretty consistently classified objects as police officers that were the same color as the uniform (e.g., blue plush toy, blue flag). This could cause the camera to follow blue cars or even the person being pulled over by the police if he/she were wearing a blue shirt. The dataset needed to be enhanced to fix this issue.

An aspect that was missing, however, was a more surefire way of determining what was not an officer. Images were added to the dataset that contained other blue items that were not labeled as a police officer. Half of the new images had both blue items and police officers and the other half had only the new items. Because the model could not just rely on color anymore to point out an officer, it had to learn some of the actual features on top of the color that signify a police officer, including the human-like shape and hat.

This required the model to be trained longer, but ultimately led to more accurate predictions overall.

B. Frame Analysis

The purpose of the frame analysis was to determine where the officer was in the image and where the camera needed to move in order to better position the officer in the frame. Each frame that the camera sent over contained metadata that housed the results of the object detection performed on the image. The object detection model would produce boxes around areas in the image that it thought were police officers. With each box was a confidence value that could be used to gauge how likely the prediction actually was to be correct. Figure 8 depicts how the frame analysis is handled to ensure the officer is located.

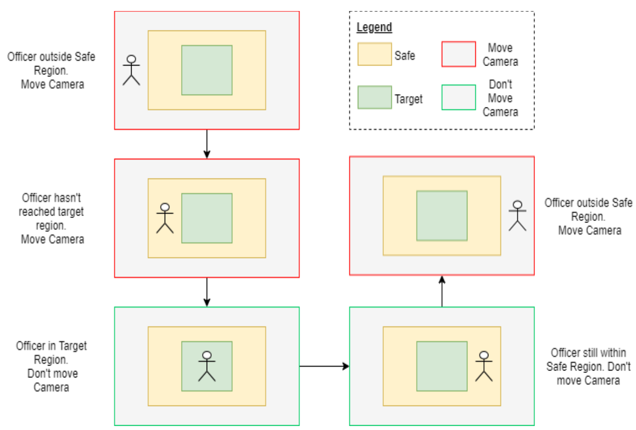


Figure 8: Visualization of interpretation of camera frames as it relates to officer location

C. Device Communications

The primary responsibility of the communication system is to administer the officer's interaction with the system. Because of the camera system's fixed location on top of the vehicle, the officer will be unable to physically interact with the camera system directly. This opens the door for the microphone device to be the bridge of communication between the officer and the system. There are a few operations within the camera system that require input from the officer to trigger. The first and most obvious is to start the camera tracking and recording of the officer. Allowing the officer to start the system will minimize the amount of wasted power and storage space consumed by the camera. The second task would be to stop tracking and recording the officer. Finally, the microphone device will need to be able to send its detected keywords over to the camera system to ensure that the keyword is recognized and processed.

The Odroid will need to communicate with both the motor FPGA on the camera system and also the microphone unit. Therefore, a command format must be created so that there is no confusion on which device is communicating with the system. The format for the commands is very simple. The first byte sent will specify which command the client is trying to send to the server. Depending on the command, the client will send additional bytes to provide supplemental information needed to process the command. In the list below, BIT 0 represents the least significant bit and BIT 7 represents the most significant bit.
Moving the camera according to the officer's location within the frame is the job of the motor control. It utilizes a UART protocol to send and receive command bytes between the ODROID and motor FPGA. Each movement involves a vertical and horizontal rotation. Two different types of movements are defined by the protocol: relative and absolute. Relative movements move the motors a specified angle from their current angle. An absolute movement will move the motors to a fixed position relative to a predefined limit switch on the gear. When resetting the system, this can be used to reposition the camera at its start location. In addition, each movement can happen synchronously or asynchronously. When performing a synchronous movement, the motors will send back a response when they reach the specified destination. An asynchronous movement will not send any response when reaching the destination. The advantage to this is that the ODROID does not have to sit and wait for a response from the motor before sending another command.

D. UART Bus Manager

UART communication is the standard interface for interconnectivity in the system. UART is used to communicate with the XBee radio that connects to the wireless microphone device, and it is used for communication with the motor FPGA. This requires an intermediary device, a Lattice MachXO2 FPGA, to facilitate multi-slave communication with the host Odroid SBC. UART does not natively support multiple slaves on one bus in its simplest implementation. To achieve this, the Lattice FPGA uses three separate channels with logic to queue messages for the host based on slave message priority and arrival time.

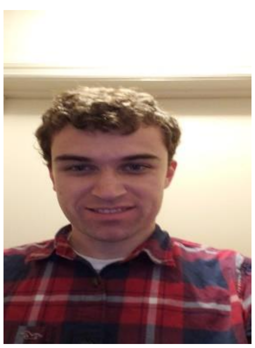
In both FPGAs, at least one UART transceiver is required to be implemented. This looks like a hardware interface with a port for writing a byte at a time as well as reading a byte at a time. The inputs to the transceiver are the RX signal, the flag clear signal, and the byte-wide transmit buffer. The outputs to the transceiver are a byte-wide receive buffer and an RX flag signal. A state machine of the logic for every received byte can be seen below. The receiver is synchronous to the system clock but counts the baud rate-specific clock pulse width of each bit by dividing the system clock by 104.

In order to reduce the error-rate of the protocol, the receiver implements oversampling on every bit reading of the state machine. This samples the RX signal line 7 times during each non-transition period and adds the sum of each sample to a register. At the 8th period, or when the bit is supposed to transition, the sum of the samples is compared. If the 7 samples equal a value greater than 3, then the saved sample is a 1. If they are less than 4, then the saved sample is 0. This averages the samples and reduces the chances of a transient or noisy signal from affecting the received data.

V. CONCLUSION

The Traffic Stop Watchdog design is definitely an idea that can be deemed as useful for the real world. Not only would our design benefit the police force, but it can also benefit civilians who are interacting with police officers. With precise stepper motor movement allowing vertical and horizontal movement, the camera will act as a set of eyes always to be tracking the police officer during an encounter. This will provide useful footage that can be used as evidence in many cases involving a police officer and a civilian. With the use of the handheld microphone device, a police officer will have direct user interface with the system and will be able to use the speech recognition technology to send keywords to the camera system.

BIOGRAPHY

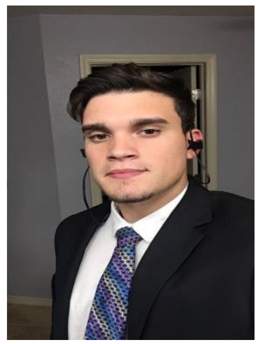


Felipe Solanet
Computer Engineering

Felipe Solanet is graduating as a Computer Engineer in the Comprehensive Track. Upon graduation, Felipe will start his career with Siemens Energy as a Software Engineer.

Michael Gendreau
Electrical Engineering

Michael Gendreau is graduating as an Electrical Engineer in the Comprehensive Track. Upon graduation, Michael will start his career with L3 Harris as a FPGA Engineer.



Jordi Niebla
Electrical Engineering

Jordi Niebla is graduating as an Electrical Engineer with the Power and Renewable Energy Track. Upon graduation, Jordi will begin his career with Florida Power & Light as an Associate Engineer in the Protection & Control department.

ACKNOWLEDGEMENTS

The authors would like to thank Dr. Richie for assisting as our Senior Design professor and providing feedback throughout the Semester. We would also like to thank Dr. Chan for his major contributions in being our advisor and guiding us throughout our journey of Senior Design 1 & 2.

We would also like to thank our committee members for taking the time out of their day to be a part of our committee and being understanding of our situation.