College of Engineering & Applied Science
ECET Senior Design

Submitted to: Professor Michael Haas
Professor Carolyn Stoll

Remotely Accessible Target System

Final Report

Chad Ruttencutter & Peter Simon

June 7, 2010

670 State Route 756
Felicity, OH 45120

June 7, 2010

Professor Michael Haas
University of Cincinnati
College of Engineering & Applied Science
2220 Victory Parkway
Cincinnati, OH 45206

Dear Professor Michael Hass:

Attached is our final report on the "Remotely Accessible Target System," which was requested by the ECET faculty.

This final report covers the details that led to the completion of our project. In this report you will see the problem, solution, credibility, and methodology. These sections will be followed by project conception, design objectives, methodology (design requirements, procedure, and testing), budget, timeline, problems encountered, analysis of problems solved, and future recommendations. The collection of these sections will illustrate the process of how the device was conceived as an idea, through its implementation.

Throughout the duration of this project, we were advised not only by you, but also Professor Elvin Stepp, Professor Laura Wilson, and Professor Carolyn Stoll. These faculty members were instrumental in the design, implementation, and documentation of our project. We appreciate your time and effort reviewing this report. If any questions or comments arise, feel free to contact either one of us.

Sincerely,

Chad A. Ruttencutter
513-678-9361
c.a.ruttencutter@gmail.com

Peter G. Simon

Peter G. Simon
513-460-7700
petergsimon@gmail.com

Department of Electrical and Computer Engineering Technology

Remotely Accessible Target System

Chad Ruttencutter and Peter Simon

June 7, 2010

**Submitted in partial fulfillment of the degree of
Bachelor of Science in
Electrical Engineering Technology**

Student Signature [illegible]

Student Signature Peter G. Simon

Advisor Signature Prof. Michael Haas

ACKNOWLEDGMENTS

We would like to thank many people for the help and support they provided during the completion of our Senior Design Project.

- Professor Michael Haas, for getting us started on the right path.
- Professor Elvin Stepp, for all the wisdom offered with hardware design.
- Professors Max Rabiee, James Everly, Kathy Ossman, Xuefu Zhou, David Tashjian, Brian Resnick, Laura Wilson, and Carolyn Stoll, for the education that made it all possible.
- Nicole Ruttencutter, for putting up with us for the last nine months.
- Our families and friends, for all the support and ideas.
- Dave Hedrick, for the bullet-proof armor.
- Irene (Prof. Haas' Aunt), for the awesome RATS logo.
- Texas Instruments, for high-quality parts at zero cost.
- Intelligrated, for the use of their plotter.
- Conard Carroll, for his sound advice in the world of C#.
- Lemos International, for quality products and excellent customer service.
- AForge.net, for image analysis libraries.
- Microsoft, IAR, Cadsoft, and Realterm for free development tools.
- Google, for more reasons than we could ever explain.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

ABSTRACT

The Remotely Accessible Target System (RATS) is a prototype that shows that a vision system, along with many other systems, can be merged to create a viable solution to nearly every problem encountered by a long distance target shooter. It is known in the shooting community that collecting data from long distance targets is dangerous and time consuming. RATS is portable, has a maximum range of 500 yards, includes a ballistic chronograph, has a resolution of less than 0.05 inches, holds at least 12 standard paper targets which can be automatically advanced, responds in less than 2 seconds, includes an easy-to-use PC interface, and retains the ability to be further expanded. All data is collected by the powerful MSP430 microcontroller, transferred via Bluetooth, processed within the software package, and then displayed to the user with a graphical user interface (GUI). All problems that were encountered were overcome. These included, but were not limited to, keystone correction, camera sleep mode, and electrical noise. Finally, many additions are being considered for the future. These improvements are as follows: a database to record all data received and processed, resizing of the form, and a range finder to verify the physical distance between the shooter and the target platform.

INTRODUCTION

The following report outlines the ten-month process involved in conceiving, conceptualizing, designing, prototyping, and completing the Remotely Accessible Target System.

The Remotely Accessible Target System (RATS), as seen in Figure 1, is a total package that allows marksmen to enjoy a safer, more leisurely shooting experience. The main objective of the system is to virtually eliminate the dangerous act of walking down range to change and/or view a target. This system also saves time by eliminating the need to manually change paper targets as well as by performing analysis and calculations, in milliseconds, that would take the average shooter several minutes.

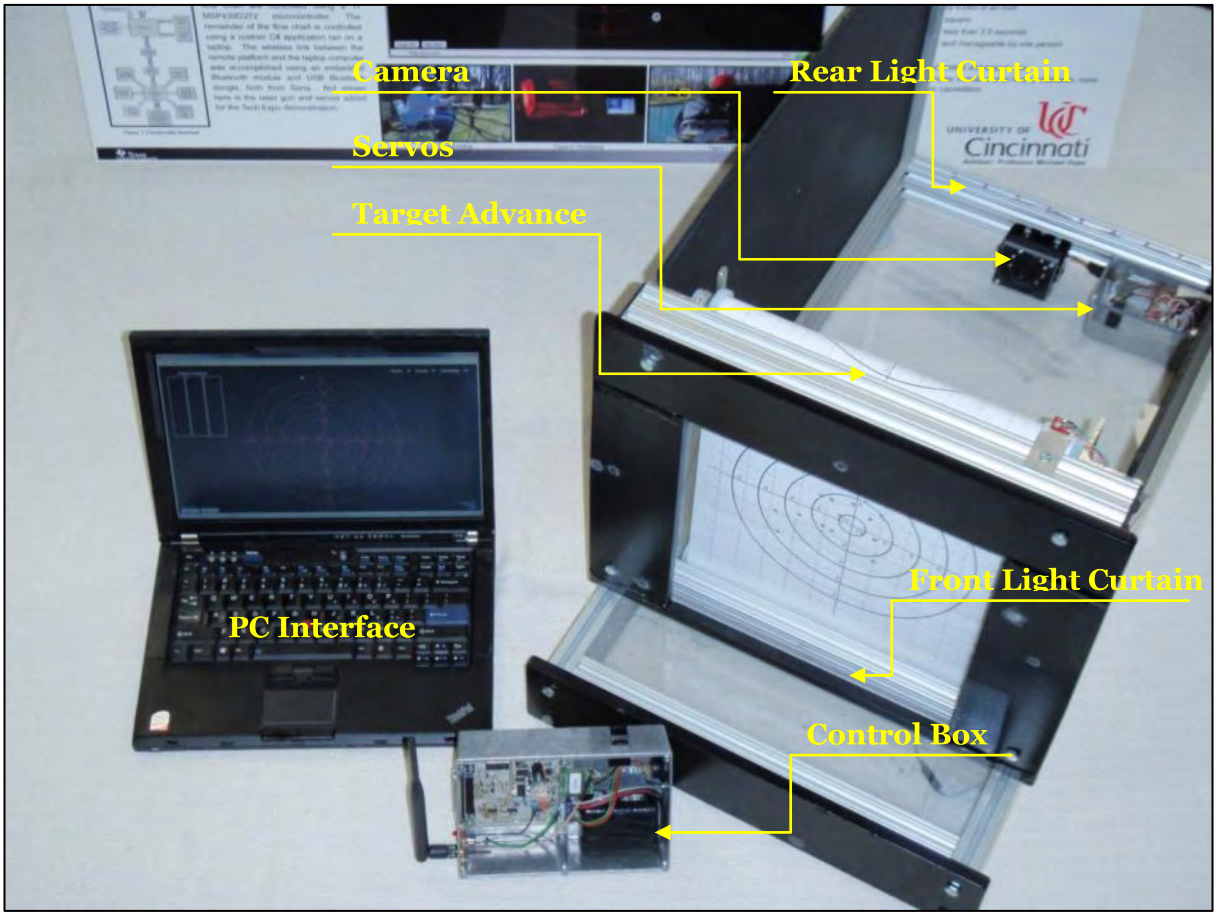


Figure 1: RATS System Overview

The system in Figure 1 is comprised of both hardware and software elements connected via a wireless link. The main objective of the software is to provide a clear and unobtrusive method of data interpretation and control. The hardware is used to collect the required data as well as to provide a platform for I/O and expansion.

In short, this is accomplished with a few main components. At the site of the target, there is a small digital camera, a microcontroller, the target advance mechanism, and a Bluetooth transceiver. At the site of the shooter, there is a Bluetooth enabled laptop, which is running our custom application. Each component has a specifically designed purpose. The combination of the microcontroller and PC command the entire operation; all other components are called on to perform their specific task. These features provide a complete system for the shooter.

More detailed system photographs are provided in Appendix B.

Problem

Shooting is potentially one of the most dangerous sports in America. While firearms only account for around 0.5% of all accidental deaths per year, this also means that about 615 Americans die yearly due to accidents with firearms [1]. The most dangerous situation encountered while participating in an organized shooting event is walking down range to replace or review a target. Often times this requires that the shooter walk between other idle shooters and the target itself. This presents the opportunity for disaster.

Not only is walking downrange a safety hazard, it is time consuming. As most shooters change his/her target every twenty shots or so, it is common that they may be spending as much time tending to the target as they are shooting. Time is a precious commodity, and any device that allows people to use their time more efficiently is a welcome improvement.

Aside from safety and time savings, another common problem with long range target shooting is the ability to clearly see shot placement on the target. Even using today's most common solution, a high power spotting scope, identifying the placement of a bullet that is smaller than the diameter of a pencil on a paper target a couple hundred yards away is extremely difficult. Not only is it hard to see one hole in a paper target, after more than about ten shots have been fired it often becomes hard to recognize which shot is the most recent. This is such a problem in long range shooting that the military often uses a spotter. This spotter is a person that sits in a pit under the target and radios or signals the shot placement to the shooter. The dangerous job of spotting could be eliminated with a system such as the Remotely Accessible Target System.

Solution

We have come up with a solution that addresses these problems, along with other common annoyances associated with target shooting. The main functionality of the Remotely Accessible Target System is two-fold. The first is to relay shot placement back to a personal computer located with the shooter via wireless communication, the final user interface as shown in Figure 2. Second, it holds 12 - 11" x 11" targets that have the ability to be changed (advanced) remotely by the user. With our target platform, we can also take speed measurements, do shot placement analysis, score keeping, single and multiplayer games, record keeping, environmental data, etc.

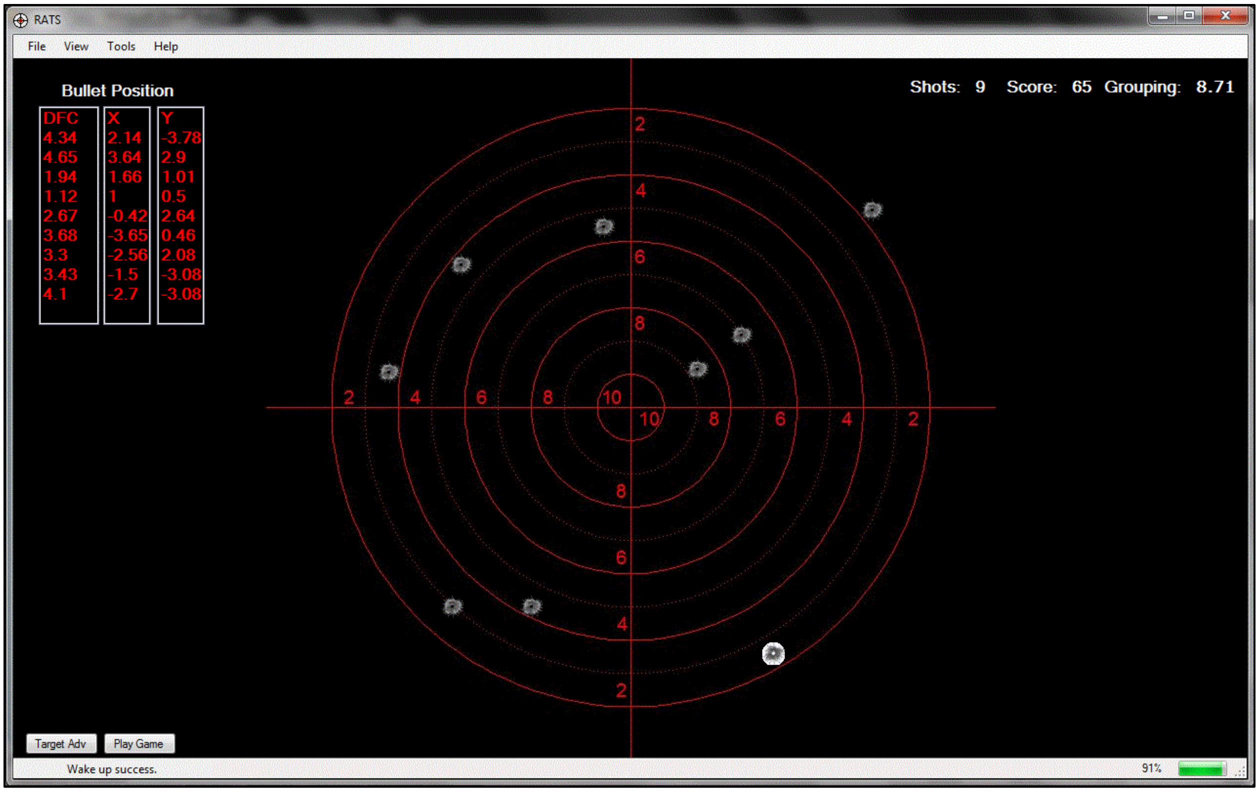


Figure 2: Remotely Accessible Target System Graphical User Interface

Figure 2 shows the main features of the user interface. In the upper left corner, a table displays the distance from the center of the target and the XY coordinates for each shot. The most recent shot is highlighted with white, and its statistics appear at the top of the table. In the upper right corner, the number of shots taken, score, and group size are shown. The bottom bar shows when events happen for instance, the camera wake up was a success. There is also a battery meter, and the chronograph readout.

The Remotely Accessible Target System uses the Texas Instruments MSP430f2272 microcontroller as its heart. This micro provides the control for all of the hardware associated with the system; such as the target advance, servo positioned laser bulls-eye, LED camera flash, chronograph, and others.

All communication is handled through a set of Bluetooth modules, one embedded and the second a USB dongle. This Bluetooth connection serves as a serial communication port between a laptop computer and the microcontroller. Using a computer, the user can control the entire target platform, from up to 500 yards away. This includes, but is not limited to, manually taking pictures, advancing targets, and performing a hard reset on the camera. A large majority of the features have also been implemented in a way that is fully automated, requiring the shooter to simply shoot.

Credibility

The collective knowledge between Chad Ruttencutter and Peter Simon spans 15 years of personal and professional experiences.

Chad Ruttencutter

Chad's primary focus was the hardware design. Chad has been tinkering in electronic and mechanical design since childhood. His interest in electronic design really started to take off after his entrance into Ford Motor Company's Electrician Apprenticeship program. This interest is what led Chad to return to college to pursue his bachelor's degree in EET, where Chad became interested in mixed signal and analog design. While at the university, Chad completed all of his co-op experience at Mitsubishi Electric, where he gained experience in vision system implementation. These experiences along with a life-long interest in shooting provided the perfect combination of experience and desire to complete this project.

Peter Simon

Peter is responsible for the creation of the user interface. Before he started at the University of Cincinnati, he had already started to learn programming in high school – Visual Basic. From there he was introduced to other languages such as Visual Basic for Applications, Intel Assembly, PCSpim, C/GCC, C++/G++, C#, MATLAB, and Allen Bradley'sRSLogix500. Not only does he have an extensive software background, but he has also completed all of the required courses to obtain a bachelor's degree in Electrical Engineering Technology. This includes courses on AC/DC circuits, digital circuits, signal processing, and much more. Though his primary purpose throughout this project was to design the software that is used by the consumer, he was also an excellent resource in the completion of the hardware design and testing.

Goals and Methodology

This project consists of two very distinct parts, software and hardware. The two primary objectives for the system are to convey to the user shot placement as well as to provide the user a platform to change paper targets automatically. The first objective is met by employing a custom-made vision system. This vision system consists of a small digital camera that records images of a specially illuminated paper target to highlight the holes in the target. This camera then relays the image to the PC via a multiplexed long range Bluetooth connection. Once received by the PC, the image is analyzed with a custom image analysis algorithm and the location of the bullet holes can be determined. The second objective is met by allowing the user to provide an input to the software interface to manually control some of the platform's features. These commands are then sent to the microcontroller via the same Bluetooth connection. The microcontroller then supplies the I/O necessary to control features such as the target advance and the servo positioned laser. This functionality, the major components, and their relationship to each other can be seen in Figure 3.

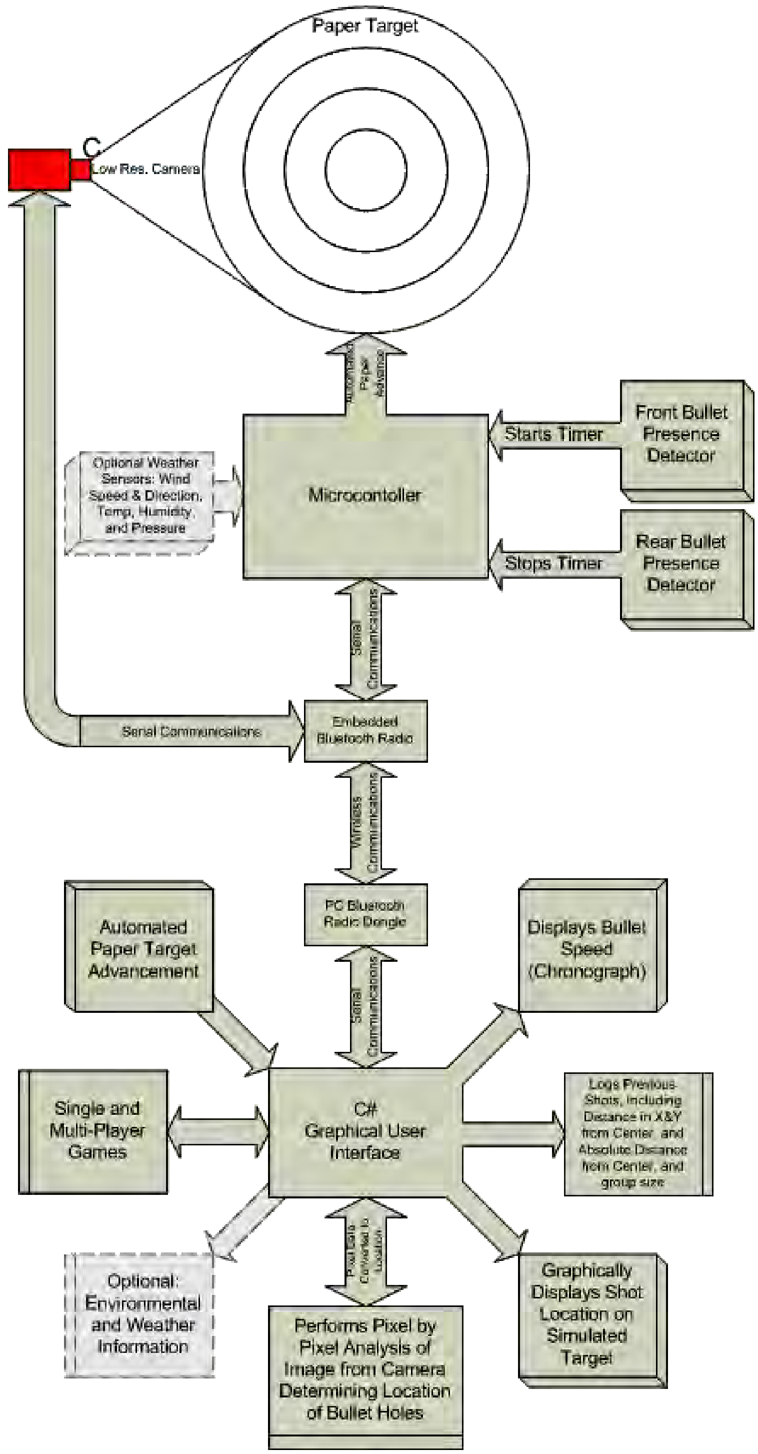


Figure 3: Block Diagram Illustrating Major Components and Functionality

The block diagram in Figure 3 depicts the different functions of the hardware, software, and communications between them. These features will be discussed in detail in the subsequent sections.

Overview

The remainder of this final report outlines in detail how the project was completed. This report includes the following sections: design objectives, technical approach, budget, timeline, problems encountered, and future recommendations.

DISCUSSION

Project Concept

The RATS project was conceptualized from a variety of previous projects and experiences. The key concept in this project was an article published in the June 2009 edition of *Nut and Volts* magazine. This article, "Ballistic Chronograph" by David Collins [3], described the process he used to build a ballistic chronograph. After reading the article, the realization was made that this bullet detecting technique could be used for much more than just measuring a bullet's velocity. Originally the use of infrared light, as discussed in the article, was planned to be used to detect the bullet's position as well as its velocity. This however quickly proved to be a difficult and complex endeavor. It was then determined to leverage both Chad's previous experience with vision inspection and Peter's expertise with software development to create a custom software package to analyze an image sent to the computer via a digital camera.

With the creation of the RATS platform, there were many opportunities to add on to the basic design. The first is an automated target advance. This quickly led to the idea of shooting at a moving target. This idea was deemed to be too difficult to do with the target advance system. This then led to a servo positioned laser pointer, where the laser became a moving bull's-eye shone on the target.

These features then led to the need for an easy-to-use software package. This custom application was intentionally modeled after a classic Window's application. This was done to ensure the average PC user could quickly learn the software and its features.

Design Objectives

The objectives and design criteria for the Remotely Accessible Target System are as follows:

Range of at least 300 yards

As the RATS is intended for use primarily by rifleman, the system had to have a relatively long wireless range. Based on our own shooting preferences, 300 yards was determined to be the minimum range acceptable. While 300 yards is the minimum, we intended to try to maintain the option of operation up to 1000 yards.

Resolution of less than 0.1 of an inch

One of the main goals for the system was to provide the user with a sense of his or her accuracy while shooting. An inherently inaccurate system would not be an acceptable solution. The maximum resolution determined to provide this level of accuracy was determined to be one hundred thousandths of an inch. The key to providing this level of accuracy is a large enough picture resolution.

Portable

Portability is the key to the usability for the RATS. This system must be manageable by one person.

Target size of at least 10 inches square

While the previous design criterion is portability, we also wanted to ensure the target was large enough to provide a pleasurable shooting experience. Based on our own shooting preferences, this minimum size was determined to be 10 inches square.

Response time of less than 3 seconds

Any system that the user deems unresponsive will eventually be cast aside. To ensure the long-term viability of the RATS, we had to ensure the system had a response time of less than three seconds. This would ensure all data would be available to the user within a three-second window of each shot.

Must hold at least 10 targets

One of the key features of the RATS is the ability to hold numerous targets. The absolute minimum number of targets to be held on the platform is 10.

Intuitive and easy to use interface

What good is any piece of electronic hardware if its interface is frustrating and hard to use? For this reason, an easy-to-use interface is a key feature that had to be designed into the RATS.

Room for expansion

To provide room for later innovation, the physical design must provide the room and capability to be expanded.

Technical Approach

The technical approach taken during the design, analysis, and testing of the Remotely Accessible Target System was in-depth and multi-faceted. The custom solutions engineered into the RATS are discussed in detail below. All system schematics are available in Appendix C, and component specific details may be found in the datasheets located in Appendix D.

Microcontroller

As mentioned earlier, every piece of hardware, with the exception of the camera, relies on the Texas Instruments MSP430f2272 microcontroller for the control necessary to make this project a success. This microcontroller exceeded all of our requirements for speed, capabilities, and peripherals. All the required programming was done in assembly language due to the need for low overhead and performance; this code is completely listed in Appendix A. The following sub-sections detail the major portions of the microcontroller's functionality.

- **Communication** – Communication is the primary role of the microcontroller. This role required the micro to receive a command from the PC using the built in Universal Serial Communication Interface in UART mode, then decipher this command to perform the function requested. In order to achieve this goal we created a four-byte command protocol which could contain all the data needed to be passed back and forth over the serial port. This command protocol is described in Table 1 below.

Table 1: RATS Command Protocol

Byte 0, Prefix	Byte 1, Function	Byte 2, Data #1	Byte 3, Data #2
0x55	0x50, ACK	Previous Cmd	Previous Data #1
	0x51, NAK	Previous Cmd	Previous Data #1
	0x52, Shot	1st byte of speed	2nd byte of speed
	0x53, Target Adv	-	-
	0x54, Move Servos	Random X pos.	Random Y pos.
	0x55, Cmd Prefix	-	-
	0x56, Change Mode	Selected Mode	Flash ON/OFF
	0x57, Time	1st byte of time	2nd byte of time
	0x58, Battery Level	1st byte of Batt Level	2nd byte of Batt Level
	0x59, Reset Camera	-	-

  The command protocol above was used by reading bytes from the serial buffer in an interrupt sub-routine. After the four bytes were read, the command was deciphered, and the proper routines were executed.

- **Bullet Velocity** – The timers of the MSP430 can run at up to the same speed as the main clock. It was this capability along with the fact that the timer values were captured without the need to execute code that led to the selection of this microcontroller. These features ensured the timer values were accurate.

The method used to measure the time the bullet took to travel from the first light curtain to the second was relatively simple. If the chronograph mode was selected, Timer A was allowed to run free. Counting from 0 to 65535, and then starting over again, when a shot was detected by the front light curtain the timer value at that moment was captured and saved in memory. When the same shot was detected by the rear light curtain, the new Timer A value was stored in memory. These two different timer values were then subtracted and the difference is the number of timer clock cycles it took the bullet to travel a fixed known distance. This value is then sent back to the PC within the last two bytes of the shot command. In the PC is where the final math is done to convert the clock cycle count into a meaningful velocity.

- **Target Advance** – This is essentially a standard stepper motor control routine with one difference. Instead of counting the steps the motor turns and then stopping, the motor only slows when this fixed count is reached. After the motor slows down, an optical limit switch is used to place the target precisely in the desired position. This optical switch looks for a black rectangle printed on the edge of the paper targets. When this black rectangle is in the proper position, the input is provided and the target advance sub-routine is completed.
- **Servo Positioned Laser** – The idea was thrown around early on about how cool it would be to somehow have a moving target to practice with. The most efficient way we could think to achieve this was to have not a moving 'target' but a moving bulls-eye on a target. This was achieved by connecting two radio control (RC) style servos together in a manner where they could then move a laser pointer in two axes. This configuration can be seen in Figure 4.

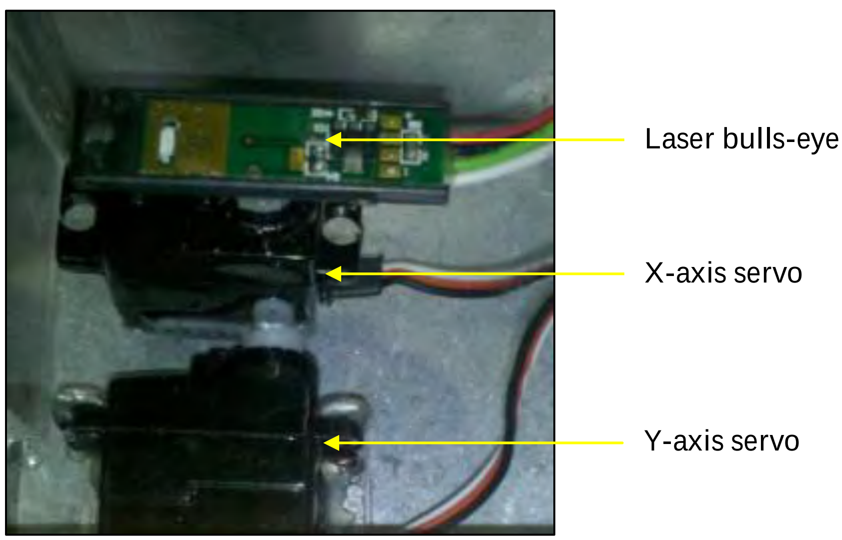


Figure 4: Servo Positioned Laser Bulls-eye

Figure 4 shows the three components to the moving bulls-eye system. The control needed to correctly position the RC servos is a 0V to 5V Pulse Width Modulation (PWM) waveform. These particular servos needed to see a PWM waveform with a high pulse ranging from 0.9ms to 2.1ms, with 1.5ms being the center position. This PWM control was easy to achieve using the MSP430's Timer B PWM outputs. This timer control allows two output pins to be set up to oscillate at a set frequency and duty factor. When the timer is started the timer controls the PWM output without any further need to control these outputs in code. This frees up the microcontroller to multitask while the servos are being positioned.

- **Battery Level** – A battery level indicator is a welcome addition to any device that is not plugged in. With the MSP430's built in Analog to Digital Converter

(ADC), monitoring the battery level of the RATS is as simple as a voltage divider and 15 lines of assembly code. The voltage divider was used to divide the battery voltage from a maximum of 8.4V down to a maximum of 3.3V. This ensured the microcontroller could read the battery voltage as a 0 to 3.3V signal, and that the higher battery voltage would not harm the microcontroller. When the microcontroller received a command to check the RATS battery it would simply run the ADC routine and convert the scaled battery level to a digital value from 0 to 1023. This value was then returned to the PC were it was converted into a relative battery life percentage.

Chronograph

The key to automating the Remotely Accessible Target System was engineering a system to reliably detect a bullet passing through the paper. To make the system capable of determining bullet velocity two identical circuits were built. The first circuit was placed in the front of the target platform, the second in the back. This hardware design provided capability for both bullet detection and bullet timing.

The method chosen to detect the bullet was a substantially improved version of David Collins Ballistic Chronograph as published in the June 2009 edition of *Nuts & Volts* magazine [3]. This design consists of three simple parts: an infrared (IR) light source, infrared photo diodes, and a filtered amplifier circuit.

The RATS bullet detection IR light source consists of 8 infrared LED emitters spaced 1.25" apart at the bottom of each chronograph light curtain. This IR light source creates a steady 'wall' of IR light at both the front and back of the RATS.

These emitters were then forward biased with a Texas Instruments REG113-5 linear voltage regulator. This regulator provided us with a small surface mount footprint at a reasonable price. One of the key features in choosing this part was the enable input. This feature allowed the regulator output to be turned off when the chronograph was not in use, reducing power consumption by approximately 30%.

With the IR light source in place, a detection circuit was needed. For this, eleven IR photo-diodes were connected in parallel and spaced 1" apart across the top of the light curtain. This parallel bank of photo-diodes was then reverse biased with the same 5V supply.

Fully illuminated by the IR emitters, this bank of reverse biased photo-diodes produces a total current of 26µA. This current was then dropped across a 56kΩ resistor producing an average DC voltage of approximately 1.5V.

As a projectile passes between the IR emitters and IR photo-diodes; the amount of infrared light detected by the photo-diodes decreases due to a shadow effect from the bullet. This decrease in IR light causes a decrease in the current being supplied by the photo-diodes. This leads to a decrease in the voltage across the 56kΩ resistor. Early testing indicated this decrease in voltage could be as small as 10mV.

The need was then to amplify a 10mV drop in voltage, with duration as short as 5µs, into a usable digital signal. Along with this performance criterion was the need to filter out any noise applied to the circuit which was outside of what was to be normally expected; mainly noise at or below 60Hz.

The final results given these design criteria can be seen in Figures 5 and 6.

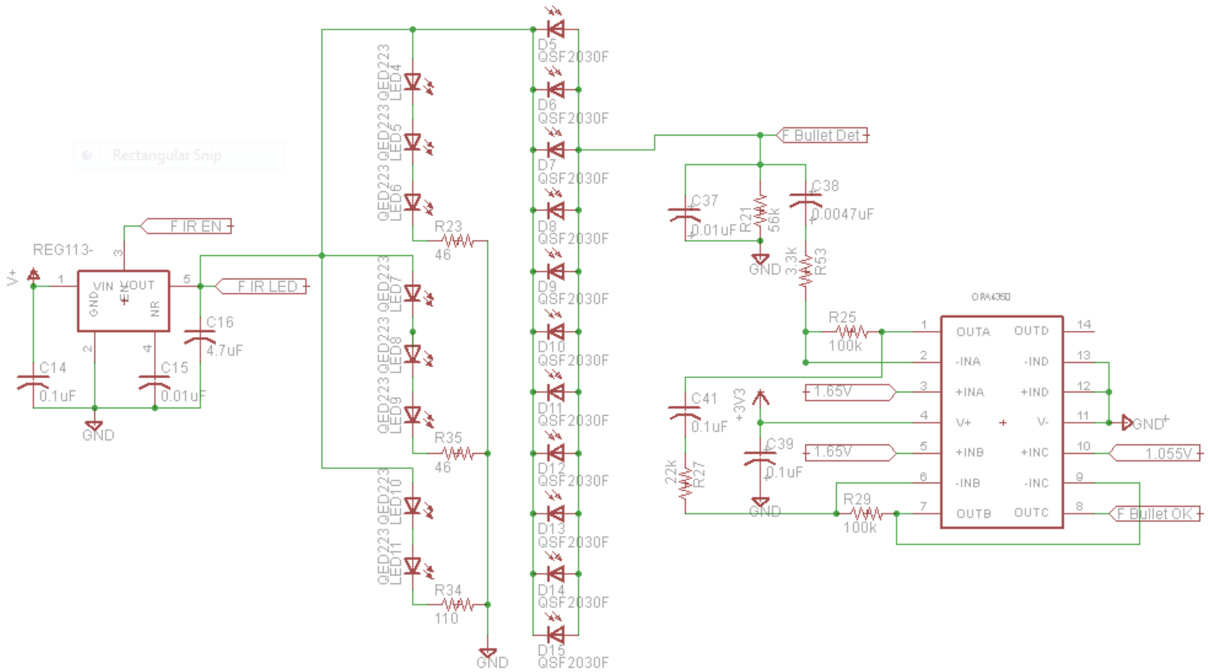


Figure 5: Chronograph Circuit Schematic

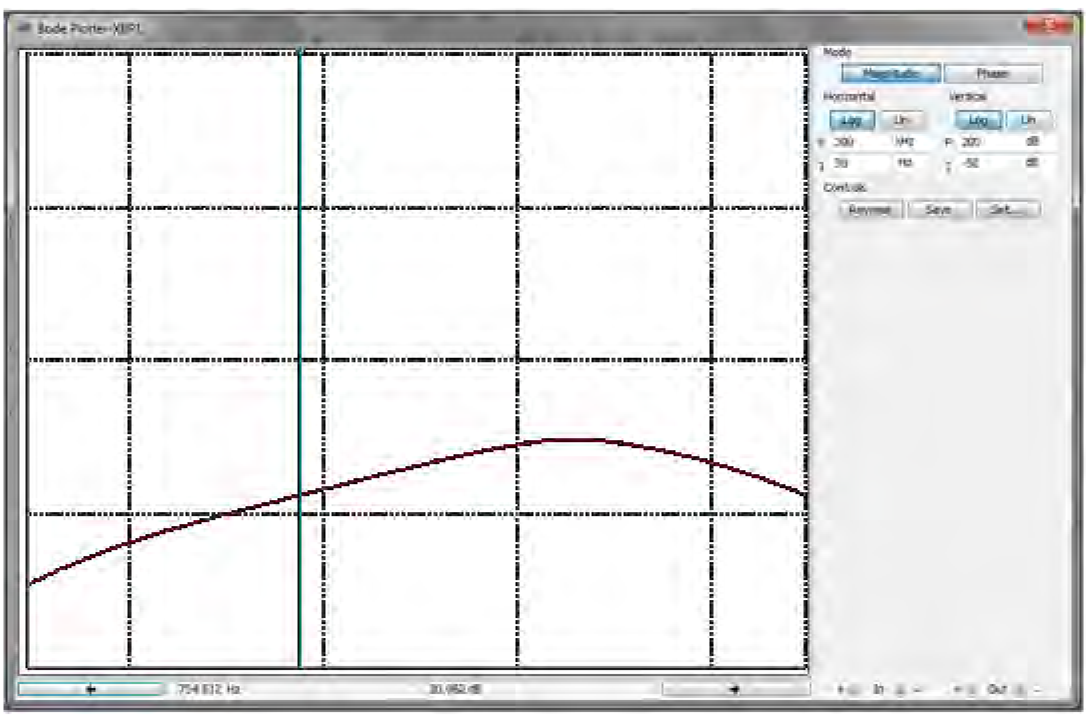


Figure 6: Bode Plot from Amplifier Circuit Simulation

As shown in Figure 5, the basic circuit for accurately detecting the voltage drop caused by a bullet is a 2-stage inverting amplifier with a built in high-pass filter followed by a voltage comparator. The simulation seen in Figure 6 shows the corner frequency of this filter at about 75Hz, which attenuates all noise at or below 90Hz.

The operational amplifier used in the circuit shown above was carefully chosen. While modeling things in this physical realm could have been done with a wide variety of Op-amps, the Texas Instruments OPA4350 was selected. This choice was made considering several key characteristics. This part's single supply and rail-to-rail performance were the primary reasons for its selection. In addition, with a slew-rate of 22V/µs, the fast performance of this part provided a nice square output from the comparator stage ensuring an accurate measurement of bullet velocity. The output of both the front and rear comparator stages can be seen in the oscilloscope screenshot in Figure 7.

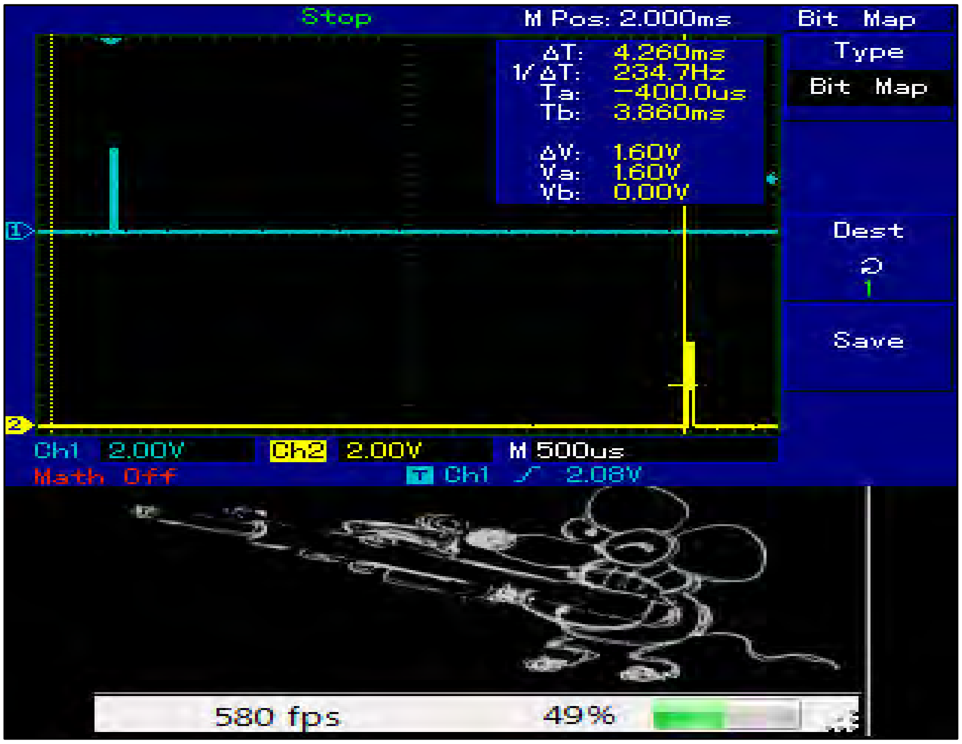


Figure 7: Oscilloscope and GUI Velocity Comparison

The oscilloscope screenshot in Figure 7 was taken while testing the light curtain and chronograph's performance. This image was taken while shooting a pellet gun through the chronograph light curtains.

With two of these circuits, finding a bullet's velocity is a relatively easy chore. The MSP430 microcontroller has two timers (A&B) that are independent of the main process. An input from each of the chronograph circuits was used to trigger the timer A capture capabilities. Upon the first bullet detection, the internal timer A value was recorded. When the second circuit detected a bullet, this timer value was then recorded again. After the second timer value was captured, the first was subtracted, giving us the number of timer counts from the first bullet detect to the second bullet detect. Dividing this value by the clock rate, and again by the distance in feet, gives us feet per second. This process can partially be seen in Figure 7. This oscilloscope screenshot in indicates a difference in time of 3.86ms from the bullet passing through the first light curtain until it passed through the second. Some quick math (0.00386s / (26.5"/12)) indicates this is equivalent to 572fps. When compared to the 580fps output by the RATS application these values indicate a percent error of 1.4% at this velocity.

Communication

Wireless communication was a mandatory part of the RATS. Due to widely available out of the box solutions, a Bluetooth wireless link was chosen to be the most practical way to establish a communications link between the target platform and the PC. One problem that had to be overcome was the need for a wireless communication link from the PC to both the MSP430 microcontroller and the C328R serial camera.

The solution to this problem was the Texas Instruments TS5A23159DGS analog switch. This integrated circuit (IC) is a double pole double throw analog switch which was used, in combination with the ESD110v2 Bluetooth module's Data Transmit Ready (DTR) output, to effectively multiplex the Bluetooth serial connection as seen in Figure 8 below.

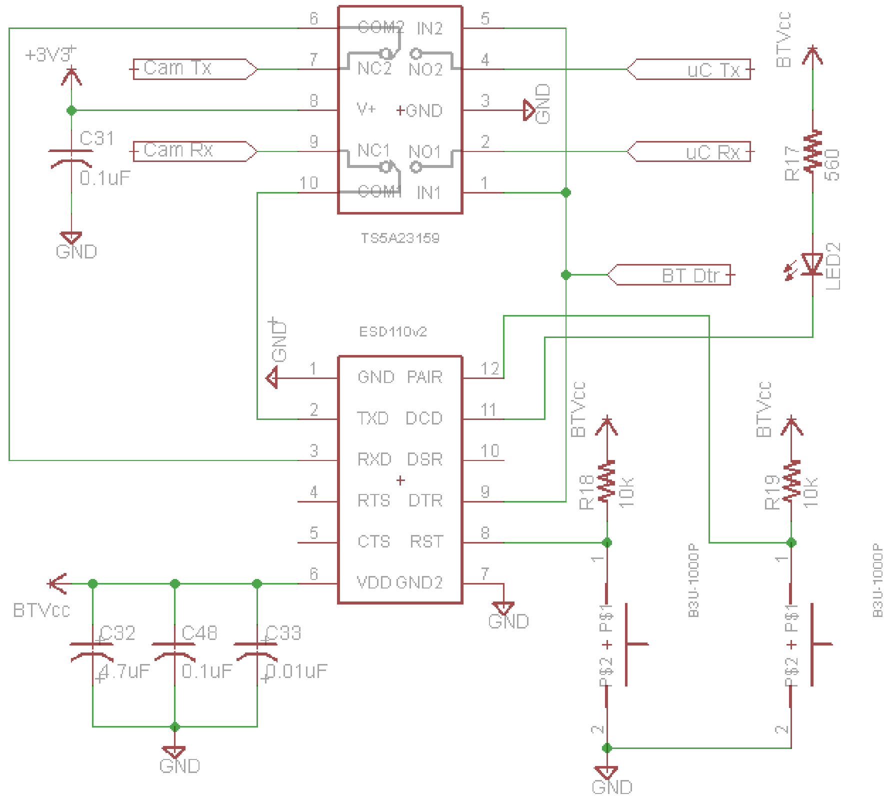


Figure 8: Multiplexed Bluetooth Wireless Connection Schematic

The schematic shown above shows how both the C328R camera (Cam Tx & Cam Rx) and the MSP430 microcontroller (uC Tx & uC Rx) are used in conjunction with the TS5A23159DGS to share one Bluetooth connection. The direction of this analog switch is controlled directly from the PC through the Bluetooth radio. The DTR output of the ESD110v2 is switched automatically by the PC from within the RATS application. The direction of the switch then depends on whether the application needs to communicate with the camera or with the MSP430. The application defines a default position to one in which the communication link is switched to the microcontroller to allow the microcontroller to send commands back to the RATS application during normal operation.

Power Regulation

There were many considerations taken into account when designing the power supply circuitry for the RATS. First and foremost was the fact that we wanted this to be a completely wireless device with a long battery life. The second most important consideration was to take into account the noise issues from the serial communication devices that we saw during early development. Another issue we chose to address while considering the first two was to replace BJT and MOSFET switches with localized switchable regulators. These solutions can be seen in Figure 9 below.

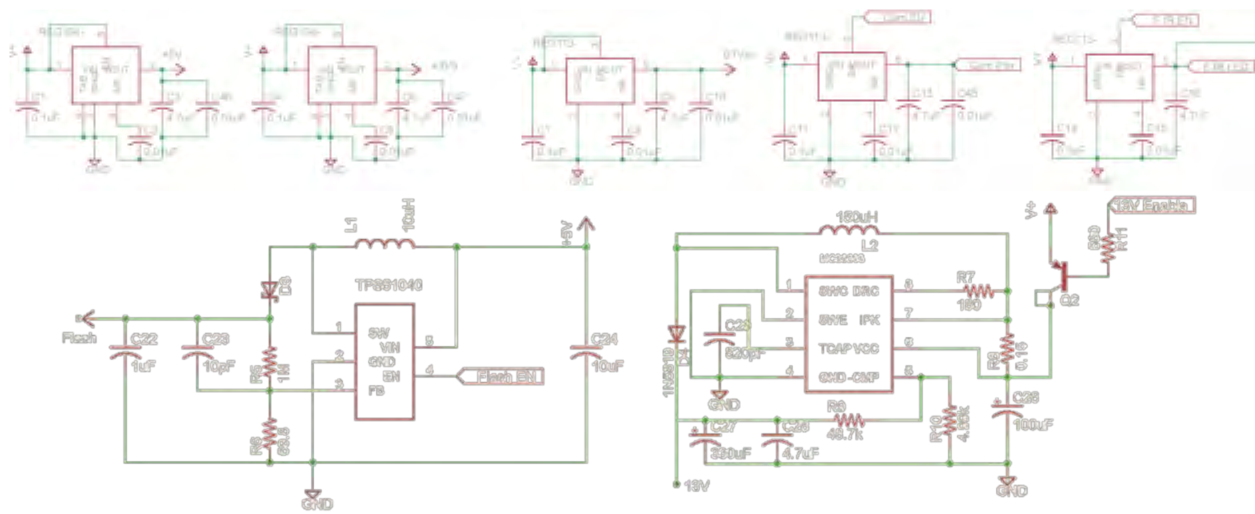


Figure 9: Power Regulation Solutions

A huge variety of voltage regulators available from Texas Instruments gave us almost unlimited options in addressing these issues. The first design issue, minimizing power consumption to maximize battery life, was addressed in many different ways.

- **Low-power Microcontroller** – Our choice of a low-power microcontroller was an easy one with the selection of the TI MSP430f2272. At our chosen operating frequency of 8MHz and voltage of 3.3V, the MSP430f2272 consumes less than 3mA of current.
- **High-efficiency Step-up DC Convertors for Low-duty Regulation** – Within the RATS platform, there are two circuits that require much higher voltages than the rest of the circuitry and are used infrequently and for brief periods of time. Using a 12V battery and regulating to 3.3V and 5V using linear regulators was deemed too inefficient. The choice was made to use the lowest voltage battery possible to power all components that are used 100% of the time. The low-duty, higher-voltage regulation would be stepped up only when needed. The high voltage circuits mentioned are the camera flash circuit and the stepper motor driver, both of which are shown in Figure 9.

  The flash circuit consists of two parallel circuits of six LEDs in series. A 15.5V regulated supply is required to power these LEDs. The regulator chosen to drive these LEDs was a Texas Instruments TPS61040. This switching regulator has a built-in enable feature that limits the quiescent current to less than 1μA when the regulator is not in use. This enable feature also reduces the need for a separate transistor switch.

  Due to size and cost constraints, the stepper motor chosen requires 12V to drive the motor coils at approximately 600mA. Another Texas Instruments step-up switching regulator, the MC33063A, was chosen to provide this power. The high current capability and ease of use drove this choice. With a rating of 1.5A it was known there would be no adverse effects due to the motors high starting current requirements.
- **Quiescent Power** - In order to limit quiescent power consumption, a 7.4V battery was used and the main circuits were regulated using Texas Instruments high-efficiency REG104 family linear regulators. These 3.3V and 5V regulators both have a current output of up to 1A, and an ultra-low drop-out voltage of less than 0.25V.

Inhibiting the noise issues experienced during prototyping was also addressed using regulation techniques. Some circuits, that were known to either impart noise onto the power bus, or to be susceptible to noise, were selected to be powered with a separate and localized regulator. These circuits are the Bluetooth transceiver, camera, and IR light curtains. Not only was it desirable to have a small package size, but an enable feature would again allow us to reduce parts count by turning load devices off and on with the regulator and not a separate switch. The regulators selected, which meet these requirements and offer a high efficiency, are the Texas Instruments REG113-5 and REG113-3.3.

Picture Analysis

The only function of the camera is to take a picture of the target paper, and relay it back to the computer. Once the picture is received, it is analyzed pixel-by-pixel. This allows for the most accurate results in finding each bullet hole. A combination of analysis techniques were used to retrieve the bullet's location. Figure 10 shows the picture as it is received by the computer.

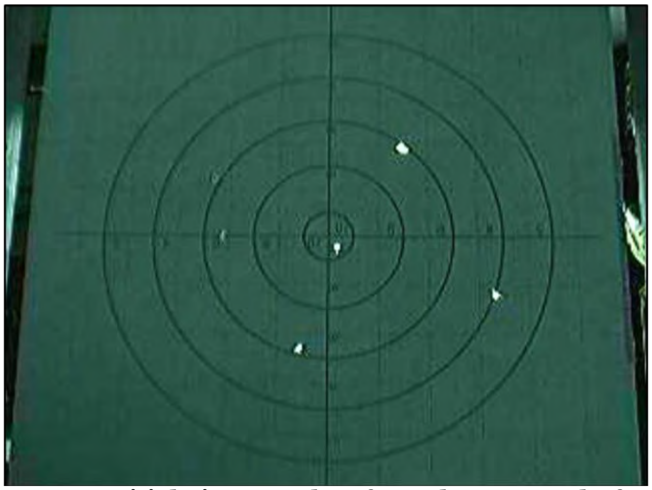


Figure 10: Initial Picture Taken from the Target Platform

In Figure 10, bullet holes are clearly visible to the eye, but they are oriented incorrectly. The first step was to reverse the image. The picture taken is from a point- of-view that is behind the paper target. If this point-of-view issue was not corrected, any shot that lands on the right side of the target would be mistakenly shown on the left. Figure 11 shows the same image after it is mirrored across the y-axis.

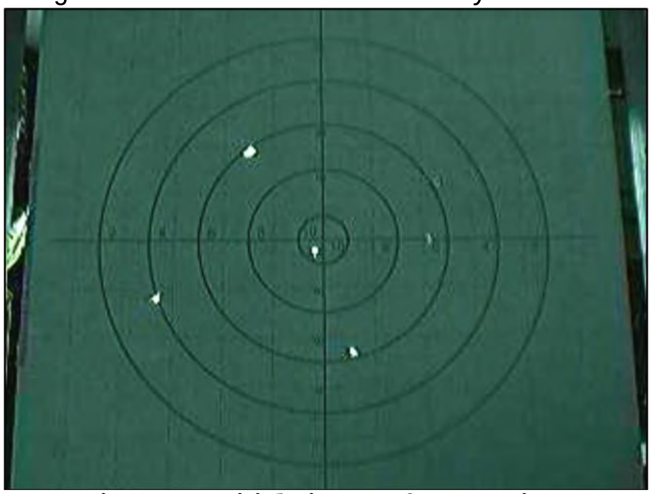


Figure 11: Initial Picture after Rotation

The mirror image in Figure 11 now shows the picture as if from the point-of-view of the shooter.

After the picture has been mirrored across the y-axis, it is turned into a black and white picture using a black and white averaging filter. This black and white image allows for a faster analysis, which we determined to be critical. Each pixel's red, green, and blue value was averaged. That corresponding value then replaced all of those RGB values. This allowed for future analysis to only be concerned with a single pixel's monochrome value, rather than each pixel's red, green, and blue values. The result of this conversion can be seen in Figure 12 below.

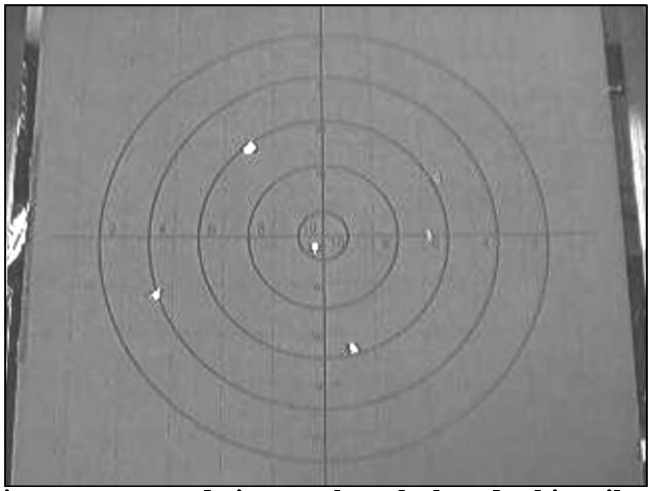


Figure 12: Rotated Picture after Black and White Filter

Following the conversion to black and white, a proprietary filtering technique was applied to the photograph. This Ruttencutter-Simon (RS) image analysis filter analyzes a black and white photograph, like the one shown above, and creates a subsequent image. This resultant image can be seen in Figure 13 below.

Figure 13: Picture after RS Filter

As seen above, the result of the RS Filter is a clearly defined image. The only information remaining in this image is the background and the blobs created using the RS filter.

Following the creation of the blobs, the remaining picture is put through a blob detection algorithm. This algorithm is used under the Lesser General Public License and was provided by AForge.net. According to AForge.NET, "AForge.NET is a C# framework designed for developers and researchers in the fields of Computer Vision and Artificial Intelligence - image processing, neural networks, genetic algorithms, machine learning, robotics, etc."[4] This library of functions was instrumental in the final analysis of our filtered picture.

Each blob found has many useful properties associated with it (area, center of gravity, fullness, X/Y position, height, and width). Within the filter a maximum and minimum height and width is set. This filters out blobs that are too small and too large. For example, in Figure 13, the blobs on the left and right side are caused from the edges of the paper target and the keystone effect. This is ignored since its size is much greater than that of a bullet.

The center of gravity property of each blob is how we determined the blob's (bullet hole's) location. When the blob detection method is called it returns the center of gravity, number of pixels contained within the blob, and the location of the blob in the image. Each center of gravity is then sent to our keystone correction algorithm.

Keystone Correction

The most challenging consideration that had to be made, in terms of software, was the keystone correction. Keystoning occurs when a picture is taken from an extreme angle. Our camera is positioned below the line of fire so it will never be hit by a bullet. This creates the kind of angle that causes the keystone effect to occur. To correct the keystone effect, an initial in-depth analysis had to be done by hand on the images.

It was observed that the relationship between a point on the bottom-left and a point on the top-right was linear. This knowledge allowed a linear correlation algorithm to be developed. First, the change in spatial density in the 'y' direction was calculated. This was important because the distance between objects appears to decrease as the height increases. The solution to this problem was finding the change in pixel density in the 'y' direction. For each center of gravity that was reported by the blob detector, the 'y' value would be scaled according to the determined change in density. This would produce the final 'y' location of the shot.

To find the 'x' position of the bullet hole, similar steps were taken, but in the 'x' direction. It can be said that points appear closer together in the 'x' direction the higher in the picture they are located. In order to correct this, the change in density in the 'x' direction also had to be calculated. In the same manner as before, the blob's center of gravity, along with the scaling factor, would produce the final 'x' location of the shot.

Flash and Sun Shade

While the above stated method of image analysis worked, it was not as reliable as needed due to the dynamic lighting conditions seen outside. The camera was later accompanied by a red LED flash. Figure 14 shows how the addition of the flash greatly enhances the clarity of each bullet hole.

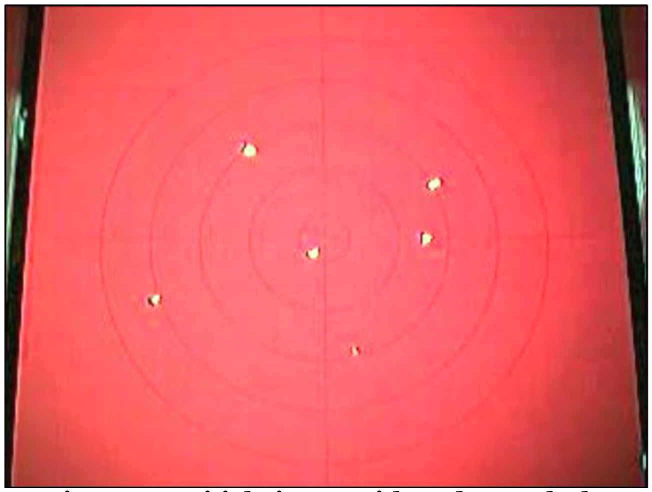


Figure 14: Initial Picture with Red LED Flash

As seen in Figure 14, this flash provided a consistent color for the camera, which made the size of the transferred picture smaller. With this flooding of light, many of the unnecessary details of the target paper are washed out, making it easier to analyze. Plastic sheathing was also added to provide shading from angled sun light. This was not initially included in our plan, but after field-testing it became very apparent that any shadows that were cast on the paper target greatly affected the quality of the picture, and became a detriment to finding shot placement.

Texas Instruments Analog Design Contest

Entering the Texas Instruments (TI) Analog Design Contest had a very positive impact in the completion of this project. TI was able to provide an integrated circuit for nearly every solution we desired. This, along with the support of the TI community, greatly improved the development cycle of the Remotely Accessible Target System. The reasons for component selection along with their impact on our project are discussed below. All system schematics are available in Appendix C, and any component specific details may be found in the datasheets located in Appendix D.

TS5A23159DGS Analog Switch

The TS5A23159DGS Analog Switch was the cornerstone of solving the problem of multiplexing three devices over one serial communications channel. This analog switch was used as a double pole double throw switch to control which device the PC was communicating with, the serial camera or the MSP430. The break-before-make feature of this IC was imperative for its use as a communications switch; this ensured no unwanted bits would be transferred to the wrong device during switching. Another important feature was the fast switching time of less than 13ns. This speed allowed the switching between communication devices to occur without incorporating any delays in the programming.

OPA4350 Operational Amplifier

A high quality Operational Amplifier (OP-AMP) was one of the most important part selections of this project. The OPA4350 exceeds our needs on every aspect. Performance and simplicity of design were the key criteria when picking this part. High performance and speed were needed as this part was not only to be used for amplification of the analog input from the light curtain, but the timing of that input also. In order to keep the design simple, we needed this performance out of a small package which provided single supply rail-to-rail operation. The OPA4350 met these criteria.

REG104 Linear Regulator

Both the 5V and 3.3V REG104 linear regulators were used in the RATS. These regulators were selected due to the need for low noise, low dropout, and small package size. The

REG104 family exceeds all of these requirements while providing fast transient regulation and high efficiency.

REG113 Linear Regulator

Both the 3.3V and 5V varieties of these parts were selected for the same reasons as were the REG104 regulators. With one added feature that set these regulators apart from others, that being the enable pin. This input pin allows the output of the regulator to be switched with the regulator itself and no external parts. This along with the small SOT23-5 package allowed for localized regulation which helped provide noise immunity for sensitive devices.

MSP430f2272 Microcontroller

The selection of the MSP430f2272 microcontroller was driven largely due to the wide variety of capability in such a small package. Texas Instruments was generous in providing us with an evaluation kit for any one of the MSP430 varieties we chose. While our original plan was to use a PIC microcontroller; after playing with the TI evaluation kit and learning the MSP430, we chose to use it instead. It turns out this was an excellent choice. The MSP430f2272 had more capability than we needed, and was very easy to use. The flexibility of the timer peripherals was key in our project due to the different timing features we needed to use. The features of the MSP430f2274 used in the RATS include: Analog to Digital Converter, Timer A capture, Timer B PWM outputs, UART serial communication, input interrupts, and others.

TPS61040 DC/DC Boost Converter

The TPS61040 DC/DC Boost Converter was used to power our camera flash. The flash circuit consisted of 12 red LEDs. Six of them in series and then in parallel with the other six in series. The voltage needed to power this configuration was 15.5V with a desired current of 75mA. This power was only needed for a duration of one second and at a relatively low frequency. It was for this reason, and the ability to quickly start the boost converter with the enable pin, that the TPS61040 was selected. Another feature that put it above others was the high switching frequency. It was estimated that any light variations at any frequency between 1kHz and 125kHz may be detected by the bullet detect circuitry as a shot. To ensure this problem would not be created with the camera flash, we chose to use a step-up regulator with as high of a switching frequency as possible. With a switching frequency of up to 1MHz, the TPS61040 was a perfect match.

BQ2057T Charge Management

The RATS uses a 5400mAh 7.4V lithium ion battery for its power source. To simplify use, it was determined an on-board battery charging circuit was needed. While looking for an optimal solution to charge the battery, there seemed to be a wide variety of commonly used battery charging circuits available. Some of these solutions were more complex than was needed. The BQ2057T from Texas Instruments was found to be the perfect compromise between performance, safety, and complexity. With features such as temperature monitoring, current and voltage control, and battery conditioning, the BQ2057t provides all the desired performance with only six external components.

MC3063A Boost Switching Regulator

The desire to keep idle power consumption to a minimum required the need for a battery which was as close to the standard operating voltage as possible. The battery selected was 7.4V. The need then came about to boost the voltage to 12V to drive the stepper motor when advancing the target. Initial testing indicated that the torque required to consistently advance the target without missing any steps was marginal at 12V. It was this reason the choice was made to drive the motor using a 13V supply instead of the rated 12 volts. This put us at 20% over maximum rated power. However, due to the low duty cycle of the motor, this was determined to not be an issue. The current draw by the motor at this voltage was calculated to be 0.38A. Therefore we desired a step up regulator that could provide this current at the required 13V. These performance criteria along with efficiency and simplicity are what drove the choice of the MC3063A switching regulator. This regulator circuit was then enabled with an external PNP transistor and coupled with a TI L293D H-Bridge to provide a complete interface between the stepper motor and the MSP430 microcontroller.

Budget

The initial budget for this project included all of the necessary components to build a working model. In Table 2, which was created in the first month of the project, all parts that we thought would be needed to create our project were included. A miscellaneous row was added to absorb all small components and extra costs. All of the parts had been thoroughly researched and specifically chosen for their tasks. This preliminary cost estimate was $440.39.

Table 2: Initial Cost Estimate

Description	Manufacturer	Part Number	Qty	Cost Per	Cost Total
MicroController	Basic Micro	Atom Nano 28	1	$8.95	$8.95
Emb. Bluetooth	Parani	Parani-ESD1000	1	$46.80	$46.80
Bluetooth Dongle	Parani	Parani-UD-100	1	$37.05	$37.05
Bluetooth Ant. 5dBi	Parani	DAT5-G01R	2	$7.95	$15.90
VGA camera	COMedia	C328-6016IR	1	$47.45	$47.45
SuperBright Red LED	Kingbright	WP7113SRD/H	6	$0.59	$3.54
7.4V 3.6Ah Li-Ion Battery	-	-	1	$20.00	$20.00
Photo Diodes	Fairchild Semiconductor	QSD2030	12	$0.22	$2.64
Differential Amplifier	Texas Instruments	Undetermined	4	$2.00	$8.00
Modular Extruded Alum.	Faztek	10EX10	16 ft	$2.80	$44.80
½ x 3" 5160 Spring Steel	McMaster-Carr	8644T43	6 ft	$9.21	$55.26
Printed Circuit Board	Undetermined	-	1	$50.00	$50.00
Misc. Hardware	N/A	N/A	-	$100.00	$100.00
				Total	$440.39

The final budget, seen in Table 3, was created in the final month of the project. This budget shows the final costs of the parts that are used in our working model. The final cost was nearly double the original estimate. The greatest contributors to this difference are structural aluminum, and plastic sheathing. The original budget allowed for 16 feet of 1-inch modular extruded aluminum. The final model required 8 feet of 1-inch, 2-inch, and 3-inch modular extruded aluminum. This larger, more rigid, extruded aluminum created a substantially more robust platform that would not have been possible using the previously budgeted 1-inch aluminum framing. No expense was spared in order for the platform to be soundly made. The plastic sheathing was a very late addition, but was also found necessary in order to provide consistent lighting for the camera. Finally, the lasers/servos were added to provide a game for the shooter.

Table 3: Final Cost Estimate

Major Components	Description	Cost
Bluetooth Communication	ESD110v2, UD100-G01, Ant.	$107.90
Serial Camera	C328R	$50.77
PCB	Custom	$52.80
Structure	Faztek Extruded Aluminum	$208.84
Enclosures	Plastic Sheathing and Boxes	$182.10
Lasers/Servo	Sparkfun	$55.90
Battery	Debco Special	$19.26
Target Advance	Motor and Custom Rollers	$38.00
Steel Armor	Custom	$55.26
IR Emitters/Receivers	Fairchild QED223, QSD2030f	$17.65
Active Components	IC's and Microcontroller	$27.83
Passive Components	Resistors, Inductors, Capacitors	$33.72
R.A.T.S.	Total	$850.03
Estimated R&D	1400 Hours @ $28/hour	$39,200.00

This final cost estimate in Table 3 also includes research and development. In the 10 months that this project was in development, over 1400 man-hours were invested. These hours included research, hardware development, software development, integration, and testing. The monetary value assigned to each hour is an average hourly wage for an entry-level engineer; this total amounted to $39,200.

There were also some costs that were not included in the budget, as they were received free of charge. Texas Instruments provided a debugger for the microcontroller that was used during the development of our project. Also, the University of Cincinnati and Microsoft's partnership provided Visual Studios. These contributions saved the team nearly $1000.

Timeline

A timeline was created in the first month. This timeline broke down the main aspects of the project, and assigned an amount of time to them. Figure 15 shows how much time was initially assigned to each task.

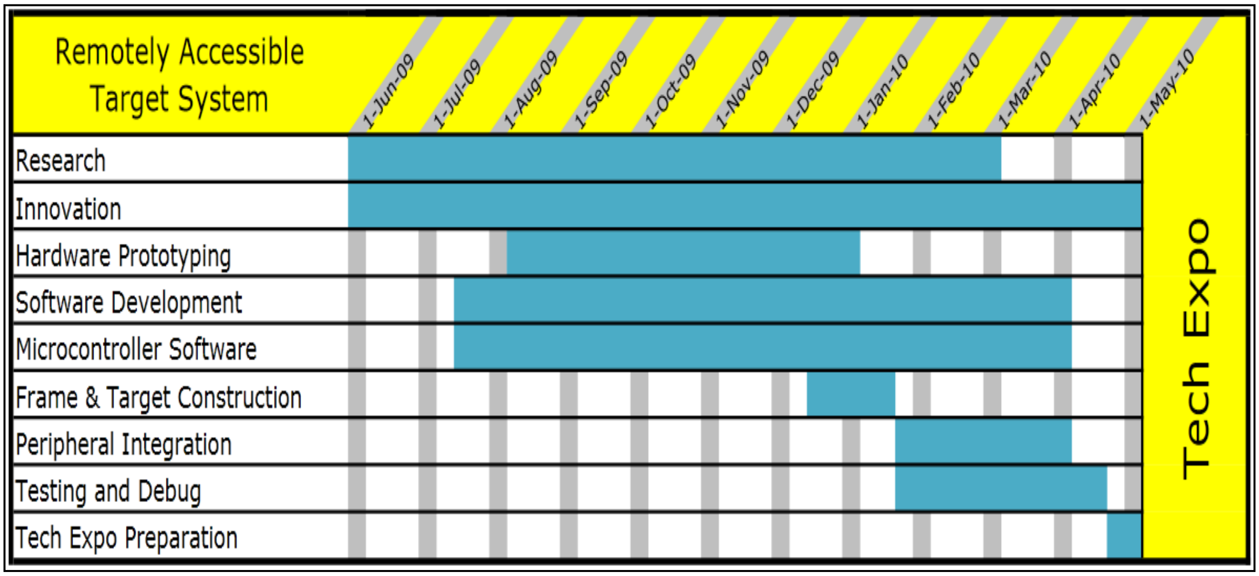


Figure 15: Initial Gantt Chart

In this original timeline, there was little room for error. The amount of work that needed to be done was immense, and is shown in the amount of time allotted. Initially the project progressed towards Demo Day. This was scheduled for the end of the winter quarter (in March) and required a working prototype. In Figure 16 the actual milestones of the project were inserted to show how closely the project followed the timeline.

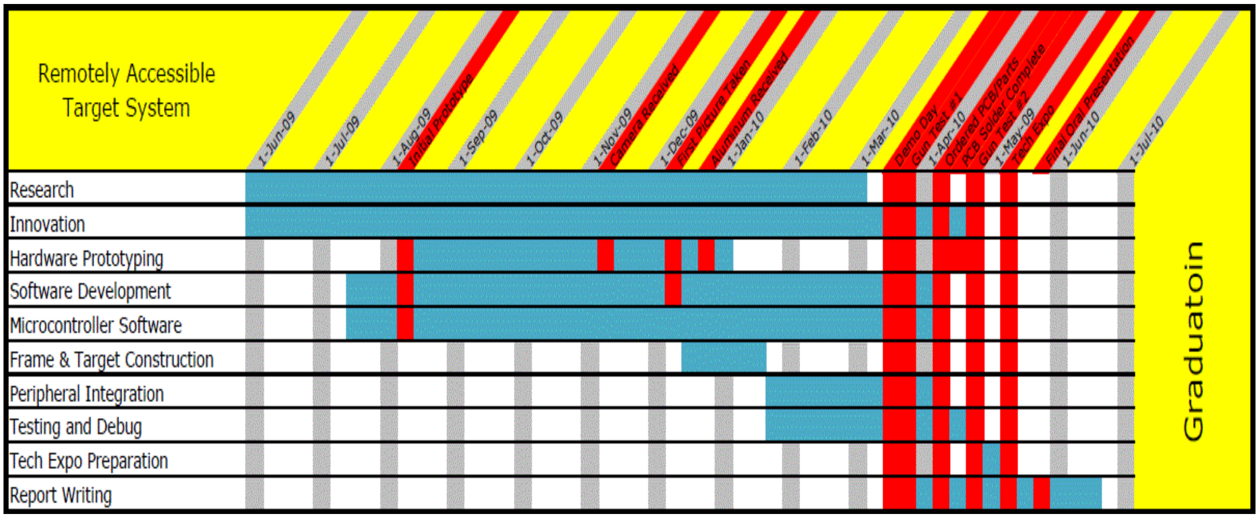


Figure 16: Final Gantt Chart

This chart shows the large milestones, and when they were achieved. This also helps show the completion of each of the major components of the target platform, and its respective user interface. All expectations were either met or exceeded for both Demo Day and Tech Expo.

Problems Encountered

Keystone Correction

The position of the camera is set below the firing line in order to never be in jeopardy of being hit by a bullet. This causes a phenomenon known in photography as the Keystone Effect. Figure 17 shows how a building, that is known to be a uniform width, can appear to become thinner as it rises farther to the sky.

Figure 17: An Example of the Keystone Effect

The picture of the building shows how the appearance is distorted by the extreme angle at which the picture is taken. This same phenomenon occurs with the target platform. In order to correct this, both the x-axis and y-axis needed to be scaled based on the distance from the bottom of the picture. This effectively warps the coordinate system we use while finding the bullet hole's location, providing us with accurate results.

Camera Sleep Mode
One of the most persistent problems during this project was a 'power saving' feature built in to the CoMedia C328R serial camera. This feature forced the camera into a sleep mode after approximately 10 seconds of idle time. After the camera enters this sleep mode, it must be synchronized and initialized again before an image can be taken. This adds approximately 2 to 3 seconds to the picture taking process and was deemed unacceptable. While this is inconvenient, the largest problem was the fact that this feature was undocumented in the camera's datasheets and manuals. This issue was only realized after the camera was deemed unsuitable and a replacement could not be found. A true solution to this problem was never found. The only thing we could do to keep the response time of the camera as short as possible was to force it to stay awake by sending it a synchronize command every seven seconds to prevent it from entering this power-saving mode.

Noise
The serial communication circuitry, which is used both in the control box and at the platform, has been found to inject noise into surrounding components and circuitry. The most detrimental part of this noise is induced onto the bullet detection circuitry from within the cable that connects the control box to the platform. This noise is only active when the PC is communicating with the camera or when the PWM outputs to the laser positioning servos are active. Fortunately, in both cases this can be ignored because a shot should not be detected at the same time the camera is relaying an image of a previous shot or when the microcontroller is positioning the servo.

Future Recommendations

Database
A database could very easily be added on to this project. Nearly all of the information that would be requested of a target shooter is already gathered, or calculated. A database would be automatically filled with information after each shot. This could include, but is not limited to, a picture of each shot, grouping, shot placement, etc. There is also available hardware space that could incorporate weather-monitoring devices. These devices would report humidity, temperature, wind speed, and wind direction. These are crucial environmental factors that can greatly affect each shot. With this information collected, a shooter can prepare for a new shooting session based on the information that was collected from previous sessions. Input boxes would be provided on the user interface so the round, gun, and sight positions can be recorded.

Off Switch Signal
Throughout testing, it was found that if the user interface was connected to the control box via Bluetooth, and then the control box's power was turned off; then an unavoidable serial error would occur. This error would crash the interface – causing unneeded wait time to the user. It was determined that it would be possible to connect a capacitor backup to the power switch of the control box. This switch would then be monitored as an interrupt input to the microcontroller. When this interrupt occurred, a command would be sent to the PC interface to disconnect the serial connection. After the switch had been powered off the capacitor would power the system for a brief period of time. This would allow for a safe power off of the control box.

Automatic Resize
Currently, the user interface must be at a fixed resolution of 1280x800. This requirement was instated due to the fact that the image of the target must remain square at all times. After the size had been locked, all the shot placements were made as an absolute distance, based on the size of the interface's form. If the target could be made to remain a square, all other placements could be set to recalculate based on the current size of the form.

Range Finder

A range finder was suggested to prove and log the distances the shooter was shooting. This would allow a competitive shooter to prove, with the pictures in the future database along with a range finder, their accuracy at a given distance. There were many suggested methods; the most prominent was a dual Global Positioning System (GPS) system. This system would integrate a GPS module into the control box, near the target platform, and a GPS module into a box near the user's laptop. These two GPS modules would relay data to the computer, which would calculate the absolute distance and log it. This feature would not only be useful to verify the distance at which rounds were shot, but can be used in conjunction with the chronograph. This would show the speed of the bullet at a certain distance, which could be used with the database to determine a projectile's deceleration.

CONCLUSION

The final outcome of this project is the working prototype shown in Figures 17 and 18. This prototype was thoroughly tested on many occasions. We feel that all of the goals, set by us and for us, were exceeded. We realize that there are areas that have room for added potential, and this will be addressed in the following months. The potential to market this model is real. With a large majority of the research behind us, the real development can begin. Many of the future recommendations are already in process, and more have been added to our personal checklists. This project may be finished with respect to the University of Cincinnati, but it is not over for us.

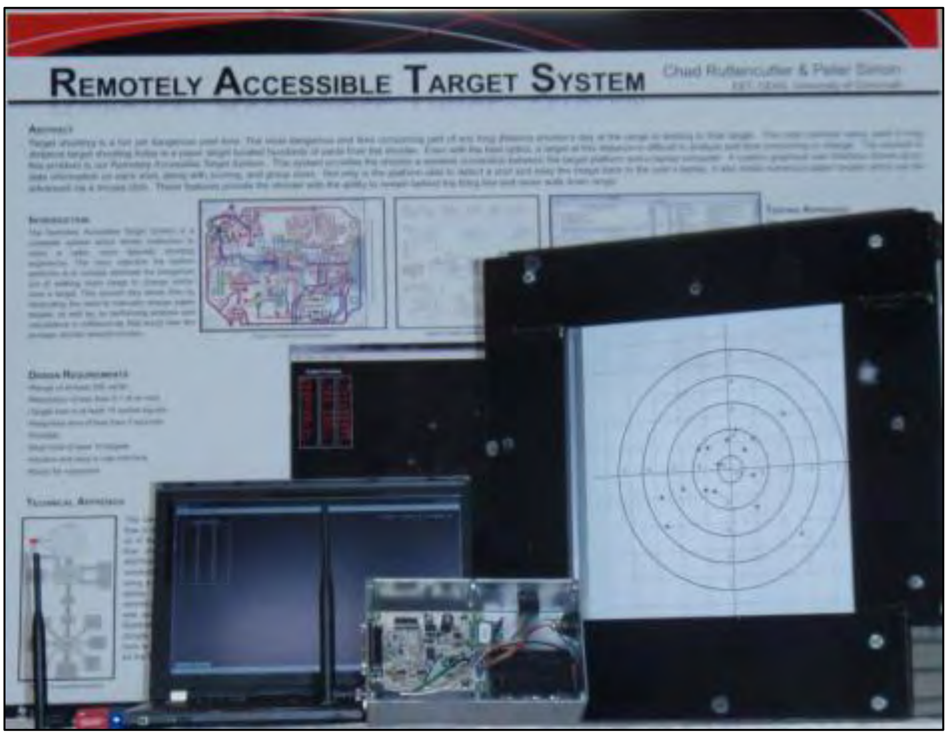


Figure 18: Remotely Accessible Target System

Figure 19: Remotely Accessible Target System

REFERENCES

[1] Institute for Legislative Action, National Rifle Association, "Firearm Safety Summary," September 23, 2008, http://www.nraila.org/Issues/factsheets/read.aspx?ID=242
[2] COMedia LTD Staff, *C328-7640 User Manual*, COMedia LTD, August 8, 2004 http://www.comedia.com.hk/FP09/Spec_PDF/C328-7640UM.pdf
[3] D. Collins, "Ballistic Chronograph" *Nuts and Volts,* pp. 36-40, June 2009.
[4] AForge.NET :: Computer Vision, Artificial Intelligence, Robotics, January, 30, 2010, http://www.aforgenet.com/

APPENDIX A

Microcontroller Code

```
//                    *****  BALLISTIC  CONCEPTS  *****
//                    Remotely Accessible Target System
//                                (RATS)
//                                 v2.6
//
//  Author: Chad Ruttencutter
//  Contributor: Peter Simon
//  uC: TI MSP430f2272
//  Date Started: January 2, 2010
//  Date Finished: May 4, 2010
//
//  Purpose: The following code was written for the RATS project.  This project
//  was conceived as a Senior Design Project for the College of Engineering &
//  Applied Science's EET Bachelor's Degree Requirement.
//
//  The purpose of the RATS target platform is to provide a long distance
//  shooter a target which detects shots, and relays the information he/she
//  desires back to a laptop computer located with the shooter.
//

#include "msp430x22x2.h"                  ; includes MSP430F22x2 specific info
;~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
;~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Assembler Setup ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
;~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
        NAME    main                      ; Defines name for entry point
        PUBLIC  main                      ; Makes the entry point visible

;~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
;~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Data ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
;~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
; Command Constants ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
CmdACK  EQU     0x50                      ; Acknowledge
CmdNAK  EQU     0x51                      ; NO Acknowledge
CmdShot EQU     0x52                      ; Shot Detected
CmdTrgt EQU     0x53                      ; Target to be advanced
CmdSrvo EQU     0x54                      ; Servo command contains pos1 and pos2
CmdPrfx EQU     0x55                      ; Command array prefix
CmdMode EQU     0x56                      ; Change Mode
CmdTme  EQU     0x57                      ; 'Time' value from timer_a capture
CmdBat  EQU     0x58                      ; Check Battery Level
CmdCam  EQU     0x59                      ; Camera Power Control, High is ON

; Defined Register Variable Names ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
        ;CPU Registers are used to reduce execution time where needed
        ;R4 primary loop counter
        ;R5 secondary loop counter
        ;R6 tertiary loop counter
        ;R7 primary interrupt loop counter
        ;R8 secondary interrupt loop counter
        ;R9 Generic Array Offset
        ;R10 initial timer_a shot detect capture value
        ;R11 secondary timer_a shot detect capture value
        ;R12 timer_a capture difference (R11-R10)
        ;R13 Command Array Offset
        ;R14 Timer_B Delay starting value
        ;R15 Temp.

#define i R4                              ; Defines Primary Loop Counter as i
#define j R5                              ; Defines Secondary Loop Counter as j
#define k R6                              ; Defines Tertiary Loop Counter as k
#define l R7                              ; Defines Primary ISR Loop Counter as l
#define m R8                              ; Defines Secondary ISR Loop Counter m
#define n R9                              ; Defines Generic Array Offset n
#define shot1 R10                         ; Defines intial shot capture val. shot1
#define shot2 R11                         ; Defines second shot capture val. shot2
#define dt  R12                           ; Defines Count between shots 1&2
#define cmd   R13                         ; Defines Command Array Offset
#define IniDly  R14                       ; Defines Timer_B intial value
#define temp  R15                         ; Defines Generic 'temp' register

; Variables ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
        ORG     0x0200                    ; Forces variables to start at 0x200
RcvCmd: DB      0x00                      ; Rcv'd command[0] should be Prefix
```

```
         DB        0x00                      ; Rcv'd command[1] Request or ACK
         DB        0x00                      ; Rcv'd command[2] Data MSB or ACK'd Cmd
         DB        0x00                      ; Rcv'd command[3] usually Data LSB

SndCmd:  DB        CmdPrfx                   ; Send command[0] always Prefix
         DB        0x00                      ; Send command[1] Request or ACK
         DB        0x00                      ; Send command[2] Data MSB or ACK'd Cmd
         DB        0x00                      ; Send command[3] usually Data LSB

Step:    DB        0x00                      ; Stepper motor step seq. Low Nibble P2
         DB        0x50                      ; Step #1
         DB        0x10                      ; Step #2
         DB        0x90                      ; Step #3
         DB        0x80                      ; Step #4
         DB        0xA0                      ; Step #5
         DB        0x20                      ; Step #6
         DB        0x60                      ; Step #7
         DB        0x40                      ; Step #8

Status:  DB        0x00                      ; Register for Status bits
         ; Bit0 = Task Complete (ie target finished, or Command finished)
         ; Bit1 = Mode 1, If (Mode 1 AND Mode 2) then Mode 3
         ; Bit2 = Mode 2, If (Mode 1 AND Mode 2) then Mode 3
         ; Bit3 = If high, camera flash on
         ; Bit4 = If high, Stepper Past Default Counts
         ; Bit5 =
         ; Bit6 =
         ; Bit7 =

         EVEN                                ; Forces next hex address to Even
TarRot:  DW        0x00C0                    ; Number of Stepping Sequences no switch
Dly2IL:  DW        0xBE00                    ; Inner Delay = 100ms long
Dly2OL:  DW        0x00                      ; Outer Delay = 100ms * Dly2OL

; Memory Address Definitions ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
         RSEG      CSTACK                    ; Defines stack location
         RSEG      CODE                      ; Defines code location

;~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
;~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ MSP430 Setup ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
;~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

RESET:   MOV       #SFE(CSTACK), SP          ; set up stack
         NOP
main:    NOP                                 ; main program
         MOV.W     #WDTPW+WDTHOLD,&WDTCTL    ; Stop watchdog timer

         BIS.B     #0x01, &P1OUT             ; Clears P1.0 => LED ON
         BIS.B     #0x01, &P1DIR             ; P1.0 Defaults to Output
         MOV.W     #0x000A, Dly2OL           ; Loads Outer Delay2 loop for ?s delay
         CALL      #TDelay2                  ; Calls Time Delay2 routine

; Clock Setup ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
         MOV.B     &CALBC1_8MHZ, &BCSCTL1    ; Set up Main Clock to run at 8Mhz
         MOV.B     &CALDCO_8MHZ, &DCOCTL     ; Set up DCO to run at 8Mhz
         BIS.B     #DIVS_1, &BCSCTL2         ; SMCLK=DCO/2=4MHz, MCLK=8MHZ

; Default Run Mode ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
         MOV.B     #0x0E, Status             ; Sets status to demo mode, flash on
         CLR.W     cmd                       ; Clears the Command Array offset
         MOV.B     #CmdPrfx, RcvCmd(R13)     ; RcvCmd[0] = CmdPrfx
         INC.W     cmd                       ; Increments Command Array offset
         MOV.B     #CmdMode, RcvCmd(R13)     ; RcvCmd[1] = CmdMode
         INC.W     cmd                       ; Increments Command Array offset
         MOV.W     #0x0103, RcvCmd(R13)      ; SndCmd[2,3] = Demo Mode, Flash on
         CALL      #RcvMode                  ; Calls Mode Command sub-routine

; I/O Pin Setup ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
     ; P1.0 #31 = Micro Indicator LED out
     ; P1.1 #32 = Capture Pin 0 in
     ; P1.2 #33 = Capture Pin 1 in
     ; P1.3 #34 = Laser Gun INTERRUPT in
```

```
; P1.4 #35 = Target Advance Limit Switch in
; P1.5 #36 =
; P1.6 #37 = Communications DTR in (Micro-controller is High)
; P1.7 #38 =

; P2.0 #8  =
; P2.1 #9  = Camera Power Control, High is ON
; P2.2 #10 = Front Light Curtain out
; P2.3 #29 = Rear Light Curtain out
; P2.4 #30 = Camera Flash Enable out
; P2.5 #3  =
; P2.6 #6  = Crystal Xin
; P2.7 #5  = Crystal Xout

; P3.0 #11 = ADC A5 battery voltage level detector
; P3.1 #12 = I2C Data line
; P3.2 #13 = I2C Serial Clock out
; P3.3 #14 =
; P3.4 #25 = UART Tx
; P3.5 #26 = UART Rx
; P3.6 #27 = Servo Laser out
; P3.7 #28 = Gun Laser out

; P4.0 #17 =
; P4.1 #18 = PWM Servo #1 output
; P4.2 #19 = PWM Servo #2 output
; P4.3 #20 = DC-DC converter stepper motor and IR Switch out
; P4.4 #21 = Stepper Motor A1 out
; P4.5 #22 = Stepper Motor B1 out
; P4.6 #23 = Stepper Motor A2 out
; P4.7 #24 = Stepper Motor B2 out

        MOV.B   #0x00, &P1OUT           ; Forces P1 Outputs and Res Pull Dir Low
        MOV.B   #0xFF, &P1DIR           ; P1 Defaults to Output
        MOV.B   #0x00, &P1SEL           ; P1 Defaults to General I/O
        BIS.B   #0x06, &P1SEL           ; P1.1 Capture Pin, P1.2 Capture Pin
        BIC.B   #0x5E, &P1DIR           ; Selects active P1 Inputs
        BIS.B   #0x5E, &P1REN           ; Activates Pull-?(~P1OUT~) Resistor
        BIS.B   #0x18, &P1OUT           ; Selected pins inputs are pulled high
        BIS.B   #0x08, &P1IES           ; P1.3 Select High to Low transition
        BIC.B   #0x08, &P1IFG           ; P1.3 IFG Cleared

        MOV.B   #0x02, &P2OUT           ; Forces P2 Outputs and Res Pull Dir Low
        MOV.B   #0xFF, &P2DIR           ; P2 Defaults to Output
        MOV.B   #0x00, &P2SEL           ; P2 Defaults to General I/O

        MOV.B   #0x00, &P3OUT           ; Forces P3 Outputs and Res Pull Dir Low
        BIS.B   #0xC0, &P3OUT           ; Sets Laser Outputs, Lasers OFF
        MOV.B   #0xFE, &P3DIR           ; P3 Defaults to Output
        MOV.B   #0x00, &P3SEL           ; P3 Defaults to General I/O
        BIS.B   #0x06, &P3SEL           ; P3.1 I2C Data, P3.2 I2C Clock out
        BIS.B   #0x30, &P3SEL           ; P3.4 UART Tx, P3.5 UART Rx

        MOV.B   #0x08, &P4OUT           ; Forces P4 Outputs and Res Pull Dir Low
        MOV.B   #0xFF, &P4DIR           ; P4 Defaults to Output
        MOV.B   #0x00, &P4SEL           ; P4 Defaults to General I/O
        BIS.B   #0x06, &P4SEL           ; Selects P4.1&2 for PWM output

; Serial Port Setup ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
        BIS.B   #UCSSEL_2, &UCA0CTL1    ; Selects SMCLK for Serial Port
//        MOV.B   #71, &UCA0BR0           ; Sets up Baud Gen. to 57600 @ 4MHz
//        MOV.B   #0, &UCA0BR1            ; Sets up Baud Gen. to 57600 @ 4MHz
//        MOV.B   #UCBRS_4,&UCA0MCTL      ; Modulation UCBRSx = 4
        MOV.B   #34, &UCA0BR0           ; Sets up Baud Gen. to 115200 @ 4MHz
        MOV.B   #0, &UCA0BR1            ; Sets up Baud Gen. to 115200 @ 4MHz
        MOV.B   #UCBRS_6, &UCA0MCTL     ; Modulation UCBRSx = 4
        BIC.B   #UCSWRST, &UCA0CTL1     ; Releases UART for operation
        BIS.B   #UCA0RXIE, &IE2         ; Enables UART Rx interrupt

; Timer_B 500kHz Timer Setup ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
        BIS.W   #TBSSEL_2+ID_3, &TBCTL  ; Timer_B: SMCLK/8 = 500kHz, Stopped
        BIC.W   #TBIE, &TBCTL           ; Clears Timer B interrupt enable
```

```
; 10 Bit ADC Conversion Setup ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
         MOV.W    #ADC10SHT_2, &ADC10CTL0  ; 16x Sample and Hold time
         BIS.B    #0x20, &ADC10AE0         ; Select P3.0 (A5) as ADC
         BIS.W    #INCH_5, &ADC10CTL1      ; Selects A5 as a single channel conv.

; Centers Laser Servos ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
         BIS.W    #TBCLR, &TBCTL           ; Clears prev. timer_b value & settings
         BIC.W    #TBIFG, &TBCTL           ; Clears Interrupt Flag
         MOV.W    #TBSSEL_2+ID_3, &TBCTL   ; Timer_B: SMCLK/8 = 500kHz, Stopped
         ; Sets up PWM Output, UP/Down Count 0~2500, Toggle/Set Points 1975~2275
         MOV.W    #0x09C4, &TBCCR0         ; Moves 2500 to TBCCR0 for 100Hz wavefrm
         MOV.W    #OUTMOD_6, &TBCCTL1      ; Selects Toggle/Set Mode
         MOV.W    #OUTMOD_6, &TBCCTL2      ; Selects Toggle/Set Mode

         MOV.W    #2042, &TBCCR1           ; Zero's Servo 1, moves duty to TBCCR1
         MOV.W    #2042, &TBCCR2           ; Zero's Servo 2, moves duty to TBCCR2
         BIS.W    #MC_3, &TBCTL            ; Starts Timer B for PWM
         MOV.W    #0x05, Dly2OL            ; Loads Outer Delay2 loop for 0.5s delay
         CALL     #TDelay2                 ; Calls Time Delay #2 routine
         BIC.W    #MC_3, &TBCTL            ; Stops Timer B for PWM

;~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
;~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Code ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
;~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

         BIC.B    #0x01, &P1OUT            ; Clears P1.0 => LED OFF
         MOV.W    #0x000A, Dly2OL          ; Loads Outer Delay2 loop for 1s delay
         CALL     #TDelay2                 ; Calls Time Delay2 routine

         BIS.B    #GIE, SR                 ; All enabled Interrupts enabled

Start:   BIC.B    #0x01, Status            ; Clears Task Complete in status reg.
         BIS.B    #0x01, &P1OUT            ; Sets P1.0 LED ON
         MOV.W    #0x0005, Dly2OL          ; Loads Outer Delay2 loop
         CALL     #TDelay2                 ; Calls Time Delay
         BIC.B    #0x01, &P1OUT            ; Clears P1.0 LED OFF
         MOV.W    #0x0005, Dly2OL          ; Loads Outer Delay2 loop
         CALL     #TDelay2                 ; Calls Time Delay

         JMP      Start                    ; Nearly Infinite Loop

;~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
;~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Subroutines ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
;~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

; Timer A Capture Shot Setup ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
CapSet:  MOV.W    #0x0260, &TACTL          ; Timer_A: SMCLK/2 = 2MHz, Cont. Mode
         MOV.W    #0x4900, &TACCTL0        ; Sets up Capture/Compare_0 Control Reg
         MOV.B    Status, temp             ; Moves Status reg to temp
         AND.B    #0x6, temp               ; Masks bit other than Mode bits
         CMP.B    #0x2, temp               ; Compares for Mode 1 Normal Operation
         JZ       NoCp1                    ; If Mode 1 rear capture is not needed
         MOV.W    #0x4900, &TACCTL1        ; Sets up Capture/Compare_1 Control Reg
NoCp1:   RET                               ; Return from sub-routine

; Flash and Wait for Camera ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
CamWait:BIT.B     #0x08, Status            ; Test Flash bit in status
         JNC      NoFlsh                   ; If not set do not turn on flash
         BIS.B    #0x10, &P2OUT            ; Sets P2.4, Turns Camera Flash ON
NoFlsh:  MOV.W    #0x19, Dly2OL            ; Loads Outer Delay for 2.0s
         CALL     #TDelay2                 ; Calls Time Delay2
         BIC.B    #0x10, &P2OUT            ; Sets P2.4, Turns Camera Flash OFF
         RET                               ; Return

; Check Received Command ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
ChkCmd:  CLR.W    cmd                      ; Clears the Cmd Array offset
         CMP.B    #CmdPrfx, RcvCmd(R13)    ; Checks for CmdPrfx
         JZ       NxtCmd0                  ; If yes, OK
         CALL     #SndNAK                  ; Else, send a NAK
NxtCmd0:INC.W     cmd                      ; Increments the Cmd Array offset
         CMP.B    #CmdACK, RcvCmd(R13)     ; Checks for CmdACK
```

```
        JNZ     NxtCmd1                  ; If not ^ then continue
        CALL    #RcvACK                  ; If so, then call sub-routine
NxtCmd1:CMP.B   #CmdNAK, RcvCmd(R13)     ; Checks for CmdNAK
        JNZ     NxtCmd2                  ; If not ^ then continue
        CALL    #RcvNAK                  ; If so, then call sub-routine
NxtCmd2:CMP.B   #CmdShot, RcvCmd(R13)    ; Checks for CmdShot
        JNZ     NxtCmd3                  ; If not ^ then continue
        CALL    #RcvShot                 ; If so, then call sub-routine
NxtCmd3:CMP.B   #CmdTrgt, RcvCmd(R13)    ; Checks for CmdTrgt
        JNZ     NxtCmd4                  ; If not ^ then continue
        CALL    #RcvTrgt                 ; If so, then call sub-routine
NxtCmd4:CMP.B   #CmdSrvo, RcvCmd(R13)    ; Checks for CmdSrvo
        JNZ     NxtCmd5                  ; If not ^ then continue
        CALL    #RcvSrvo                 ; If so, then call sub-routine
NxtCmd5:CMP.B   #CmdMode, RcvCmd(R13)    ; Checks for CmdMode
        JNZ     NxtCmd6                  ; If not ^ then continue
        CALL    #RcvMode                 ; If so, then call sub-routine
NxtCmd6:CMP.B   #CmdTme, RcvCmd(R13)     ; Checks for CmdTme
        JNZ     NxtCmd7                  ; If not ^ then continue
        CALL    #RcvTme                  ; If so, then call sub-routine
NxtCmd7:CMP.B   #CmdBat, RcvCmd(R13)     ; Checks for CmdBat
        JNZ     NxtCmd8                  ; If not ^ then continue
        CALL    #RcvBat                  ; If so, then call sub-routine
NxtCmd8:CMP.B   #CmdCam, RcvCmd(R13)     ; Checks for Camera Reset Cmd
        JNZ     NxtCmdF                  ; If not ^ then continue
        CALL    #RcvCam                  ; If so, then call sub-routine
NxtCmdF:BIT.B   #0x01, Status            ; Test Status for Task Complete
        JC      CmdComp                  ; If Task Complete, Cmd Complete
        CALL    #SndNAK                  ; If no Cmd's confirmed or Task Not Comp
        JMP     CmdCmp2                  ; Then send NAK and no ACK
CmdComp:CALL    #SndACK                  ; If Task completed send ACK
CmdCmp2:BIC.B   #0x01, Status            ; Clears the Task Complete Flag
        RET                              ; Return

; Demo Shot Detected ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
DmoSht: MOV.W   R10, R12                 ; Moves shot1 (R10) to R12 for sub
        SUB.W   R11, R12                 ; Subtracts shot2 (R11) from shot1
        SWPB    dt                       ; Swaps bytes of Shot Time for SndCmd
        CLR.W   cmd                      ; Clears the Command Array offset
        MOV.B   #CmdPrfx, SndCmd(R13)    ; SndCmd[0] = CmdPrfx
        INC.W   cmd                      ; Increments Command Array offset
        MOV.B   #CmdShot, SndCmd(R13)    ; SndCmd[1] = CmdShot
        INC.W   cmd                      ; Increments Command Array offset
        MOV.W   R12, SndCmd(R13)         ; SndCmd{2,3} = Shot Time (dt)
        BIC.B   #0x80, &P3OUT            ; Clears P3.8, Laser Gun laser ON
        CALL    #TxCmd                   ; Calls Transmit Command sub-routine
        CALL    #CamWait                 ; Calls the Camera Wait and Flash Rtne
        BIS.B   #0x80, &P3OUT            ; Sets P3.8, Laser Gun laser OFF
        RET                              ; Return

; Received ACK Command ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
RcvACK: BIT.B   #0x08, Status            ; Tests for camera flash status bit set
        JNC     MkFlsh                   ; If it is not set, make a flash
        CALL    #CamWait                 ; Otherwise Flash for Comm. Test
        JMP     AckDun                   ; Then finish Ack
MkFlsh: BIS.B   #0x08, Status            ; Set the Flash enable bit
        CALL    #CamWait                 ; Fires Camera Flash for Comm. Test
        BIC.B   #0x08, Status            ; Flash EN was clr to start so clear
AckDun: BIS.B   #0x01, Status            ; Sets Task Complete Bit in Status
        RET                              ; Return

; Received Check Battery Command ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
RcvBat: BIS.W   #ADC10ON, &ADC10CTL0     ; ADC10 ON
        BIS.W   #ENC+ADC10SC, &ADC10CTL0; Starts sampling and Conversion
ChkADC: BIT.W   #ADC10IFG, &ADC10CTL0    ; Tests for end of conversion
        JNC     ChkADC                   ; If Conversion not finished, loop
        MOV.W   &ADC10MEM, temp          ; Moves the ADC Value to temp
        SWPB    temp                     ; Swaps Bytes for transmission
        BIC.W   #ADC10IFG, &ADC10CTL0    ; Clears the end of conv flag
        CLR.W   cmd                      ; Clears the Command Array offset
        MOV.B   #CmdPrfx, SndCmd(R13)    ; SndCmd[0] = CmdPrfx
        INC.W   cmd                      ; Increments Command Array offset
```

```
        MOV.B   #CmdBat, SndCmd(R13)      ; SndCmd[1] = CmdBat
        INC.W   cmd                       ; Increments Command Array offset
        MOV.W   R15, SndCmd(R13)          ; SndCmd[2,3} = ADC Battery Level (R15)
        CALL    #TxCmd                    ; Calls Transmit Command sub-routine
        BIS.B   #0x01, Status             ; Sets Task Complete Bit in Status
        RET                               ; Return

; Received Camera Reset Command ------------------------------------------------
RcvCam: BIC.B   #0x02, &P2OUT             ; Turns off power to camera
        MOV.W   #0x64, Dly2OL             ; Loads Outer Delay2 loop for 10s delay
        CALL    #TDelay2                  ; Calls Time Delay #2 routine
        BIS.B   #0x02, &P2OUT             ; After delay, turns camera back on
        BIS.B   #0x01, Status             ; Sets Task Complete Bit in Status
        RET                               ; Return

; Received NAK Command ---------------------------------------------------------
RcvNAK: CALL    #TxCmd                    ; Re-transmits the last sent command
        RET                               ; Return

; Received Shot Detected Command -----------------------------------------------
RcvShot:BIS.B   #0x01, Status             ; Sets Task Complete Bit in Status
        RET                               ; Return

; Received Servo Command --------------------------------------------------------
RcvSrvo:BIS.W   #TBCLR, &TBCTL            ; Clears prev. timer_b value & settings
        BIC.W   #TBIFG, &TBCTL            ; Clears Interrupt Flag
        MOV.W   #TBSSEL_2+ID_3, &TBCTL    ; Timer_B: SMCLK/8 = 500kHz, Stopped
        ; Sets up PWM Output, UP/Down Count 0~2500, Toggle/Set Points 1975~2275
        MOV.W   #0x09C4, &TBCCR0          ; Moves 2500 to TBCCR0 for 100Hz wavefrm
        MOV.W   #OUTMOD_6, &TBCCTL1       ; Selects Toggle/Set Mode
        MOV.W   #OUTMOD_6, &TBCCTL2       ; Selects Toggle/Set Mode
        ; 'Zeros' Servo Position before moving to increase stability
        MOV.W   #2042, &TBCCR1            ; Zero's Servo 1, moves duty to TBCCR1
        MOV.W   #2042, &TBCCR2            ; Zero's Servo 2, moves duty to TBCCR2
        BIS.W   #MC_3, &TBCTL             ; Starts Timer B for PWM
        MOV.W   #0x05, Dly2OL             ; Loads Outer Delay2 loop for 0.5s delay
        CALL    #TDelay2                  ; Calls Time Delay #2 routine
        BIC.W   #MC_3, &TBCTL             ; Stops Timer B for PWM
        INC.W   cmd                       ; Increments the Cmd Array offset
        CLR.W   temp                      ; Clears the (word) temp Reg (R15)
        MOV.B   RcvCmd(R13), temp         ; Moves the Rcv'd duty factor to temp
        ADD.W   #1997, temp               ; Adds 1997 offset to the rcv'd value
        MOV.W   R15, &TBCCR1              ; Moves Servo #1 duty factor to TBCCR1
        INC.W   cmd                       ; Increments the Cmd Array offset
        CLR.W   temp                      ; Clears the (word) temp Reg (R15)
        MOV.B   RcvCmd(R13), temp         ; Moves Rcv'd duty factor to temp
        ADD.W   #1997, temp               ; Adds 1997 offset to the rcv'd value
        MOV.W   R15, &TBCCR2              ; Moves Servo #2 duty factor to TBCCR2
        BIS.W   #MC_3, &TBCTL             ; Starts Timer B for PWM
        MOV.W   #0x05, Dly2OL             ; Loads Outer Delay2 loop for 0.5s delay
        CALL    #TDelay2                  ; Calls Time Delay #2 routine
        BIC.W   #MC_3, &TBCTL             ; Stops Timer B for PWM
        MOV.W   #0x01, Dly2OL             ; Loads Outer Delay2 loop for 0.1s delay
        CALL    #TDelay2                  ; Calls Time Delay #2 routine
        BIC.W   #0x10, &TACCTL0           ; Clears Capture 0 interrupt enable
        BIC.W   #0x10, &TACCTL1           ; Clears Capture 1 interrupt enable
        BIC.B   #0x08, &P1IFG             ; P1.3 IFG Cleared
        MOV.W   #0x0000, dt               ; Moves shot1 (R10) to R12 for sub
        BIS.B   #GIE, SR                  ; All enabled Interrupts enabled
        BIC.B   #0x40, &P3OUT             ; Clrs P3.6 Servo Laser ON
        MOV.W   #0x0014, Dly2OL           ; Loads Outer Delay2 loop for 2s delay
        MOV.W   Dly2OL, shot1             ; Moves delay value to shot1
        CALL    #TDelay2                  ; Calls Time Delay #2 routine
        BIS.B   #0x40, &P3OUT             ; Sets P3.6 Servo Laser OFF
        BIC.B   #GIE, SR                  ; All enabled Interrupts disabled
        BIS.B   #0x01, Status             ; Sets Task Complete Bit in Status

        RET                               ; Return

; Received Advance Target Command ----------------------------------------------
//
//---- Lookup Table would only work in debug. No solution found, so lookup table
```

```
//~~~~ was eliminated with a lot of repetitive code.  TI would not provide
//~~~~ support.
//RcvTrgt:
//BIC.B    #0x20, &P2OUT            ; Enables DC-DC Converter & Limit Switch
//BIC.B    #0x08, &P4OUT            ; Enables DC-DC Converter & Limit Switch
//Turn:    MOV.W    #0x08, n                 ; Loads the Array offset
//Steps8:  BIC.B    #0xF0, &P4OUT            ; Clears lower nibble of P2
//         BIS.B    Step(R9), &P4OUT         ; Sets bits for current step phase
//         MOV.W    #0xF000, IniDly          ; Loads the Timer_B intial count (35ms)
//         CALL     #TDelay                  ; Calls Time Delay Sub-routine
//         DEC.W    n                        ; Decrements phase loop counter
//         JNZ      Steps8                   ; If Phase not zero, then go again
//         DEC.W    TarRot                   ; If Fin, then Dec. Target Rotate Count
//         JNZ      Turn                     ; If not zero turn more
//NotDun:  MOV.W    #0x08, n                 ; Loads the Array offset
//Steps82:BIC.B     #0xF0, &P4OUT            ; Clears lower nibble of P2
//         BIS.B    Step(R9), &P4OUT         ; Sets bits for current step phase
//         MOV.W    #0xC000, IniDly          ; Loads the Timer_B intial count (50ms)
//         CALL     #TDelay                  ; Calls Time Delay Sub-routine
//         BIT.B    #0x10, &P1IN             ; Tests P1.4 Target Advance Limit
//         JC       TarFin                   ; If high Target Advance is Complete
//         DEC.W    n                        ; Decrements phase loop counter
//         JNZ      Steps82                  ; If Phase not zero, then go again
//         JMP      NotDun                   ; Target Advance Limit is not made cont
//TarFin:  MOV.W    #0x00C0, TarRot          ; When Finished restore Rotate Count
//         BIC.B    #0xF0, &P4OUT            ; De-energizes motor coils
//         BIS.B    #0x20, &P2OUT            ; Disables DC-DC Conv. & Limit Switch
//         BIS.B    #0x01, Status            ; Sets Task Complete Bit in Status
//
//~~~~ Lookup Table would only work in debug. No solution found, so lookup table
//~~~~ was eliminated with a lot of repetitive code.  TI would not provide
//~~~~ support.
//
RcvTrgt:BIC.B    #0x08, &P4OUT          ; Enables DC-DC Converter & Limit Switch
        MOV.W    #0x00C0, TarRot        ; Loads Default Rotation Count
        MOV.W    #0xF000, R6            ; Loads 0xF000 into register for delay
        BIC.B    #0x10, Status          ; Clears Task complete bit in status
Step1:  MOV.B    &P4OUT, temp           ; Moves current P4 outputs into temp
        BIC.B    #0xF0, temp            ; Clears out top nibble of temp P4
        BIS.B    #0x40, temp            ; Sets the current step bits in temp
        MOV.B    R15, &P4OUT            ; Moves the new value of P4 to P4out
        MOV.W    R6, R14                ; Loads the Timer_B intial count (35ms)
        CALL     #TDelay                ; Calls Time Delay Sub-routine
        BIT.B    #0x10, Status          ; Tests for initial rotation complete
        JNC      Step2                  ; If initial rotation N/comp cont.
        BIT.B    #0x10, &P1IN           ; If initial rotat. comp. test lmt swtch
        JC       TarFin                 ; If lmt switch is set; Target finished
Step2:  MOV.B    &P4OUT, temp           ; Moves current P4 outputs into temp
        BIC.B    #0xF0, temp            ; Clears out top nibble of temp P4
        BIS.B    #0x60, temp            ; Sets the current step bits in temp
        MOV.B    R15, &P4OUT            ; Moves the new value of P4 to P4out
        MOV.W    R6, R14                ; Loads the Timer_B intial count (35ms)
        CALL     #TDelay                ; Calls Time Delay Sub-routine
        BIT.B    #0x10, Status          ; Tests for initial rotation complete
        JNC      Step3                  ; If initial rotation N/comp cont.
        BIT.B    #0x10, &P1IN           ; If initial rotat. comp. test lmt swtch
        JC       TarFin                 ; If lmt switch is set; Target finished
Step3:  MOV.B    &P4OUT, temp           ; Moves current P4 outputs into temp
        BIC.B    #0xF0, temp            ; Clears out top nibble of temp P4
        BIS.B    #0x20, temp            ; Sets the current step bits in temp
        MOV.B    R15, &P4OUT            ; Moves the new value of P4 to P4out
        MOV.W    R6, R14                ; Loads the Timer_B intial count (35ms)
        CALL     #TDelay                ; Calls Time Delay Sub-routine
        BIT.B    #0x10, Status          ; Tests for initial rotation complete
        JNC      Step4                  ; If initial rotation N/comp cont.
        BIT.B    #0x10, &P1IN           ; If initial rotat. comp. test lmt swtch
        JC       TarFin                 ; If lmt switch is set; Target finished
Step4:  MOV.B    &P4OUT, temp           ; Moves current P4 outputs into temp
        BIC.B    #0xF0, temp            ; Clears out top nibble of temp P4
        BIS.B    #0xA0, temp            ; Sets the current step bits in temp
        MOV.B    R15, &P4OUT            ; Moves the new value of P4 to P4out
        MOV.W    R6, R14                ; Loads the Timer_B intial count (35ms)
```

```
          CALL      #TDelay                       ; Calls Time Delay Sub-routine
          BIT.B     #0x10, Status                 ; Tests for initial rotation complete
          JNC       Step5                         ; If initial rotation N/comp cont.
          BIT.B     #0x10, &P1IN                  ; If initial rotat. comp. test lmt swtch
          JC        TarFin                        ; If lmt switch is set; Target finished
Step5:    MOV.B     &P4OUT, temp                  ; Moves current P4 outputs into temp
          BIC.B     #0xF0, temp                   ; Clears out top nibble of temp P4
          BIS.B     #0x80, temp                   ; Sets the current step bits in temp
          MOV.B     R15, &P4OUT                   ; Moves the new value of P4 to P4out
          MOV.W     R6, R14                       ; Loads the Timer_B intial count (35ms)
          CALL      #TDelay                       ; Calls Time Delay Sub-routine
          BIT.B     #0x10, Status                 ; Tests for initial rotation complete
          JNC       Step6                         ; If initial rotation N/comp cont.
          BIT.B     #0x10, &P1IN                  ; If initial rotat. comp. test lmt swtch
          JC        TarFin                        ; If lmt switch is set; Target finished
Step6:    MOV.B     &P4OUT, temp                  ; Moves current P4 outputs into temp
          BIC.B     #0xF0, temp                   ; Clears out top nibble of temp P4
          BIS.B     #0x90, temp                   ; Sets the current step bits in temp
          MOV.B     R15, &P4OUT                   ; Moves the new value of P4 to P4out
          MOV.W     R6, R14                       ; Loads the Timer_B intial count (35ms)
          CALL      #TDelay                       ; Calls Time Delay Sub-routine
          BIT.B     #0x10, Status                 ; Tests for initial rotation complete
          JNC       Step7                         ; If initial rotation N/comp cont.
          BIT.B     #0x10, &P1IN                  ; If initial rotat. comp. test lmt swtch
          JC        TarFin                        ; If lmt switch is set; Target finished
Step7:    MOV.B     &P4OUT, temp                  ; Moves current P4 outputs into temp
          BIC.B     #0xF0, temp                   ; Clears out top nibble of temp P4
          BIS.B     #0x10, temp                   ; Sets the current step bits in temp
          MOV.B     R15, &P4OUT                   ; Moves the new value of P4 to P4out
          MOV.W     R6, R14                       ; Loads the Timer_B intial count (35ms)
          CALL      #TDelay                       ; Calls Time Delay Sub-routine
          BIT.B     #0x10, Status                 ; Tests for initial rotation complete
          JNC       Step8                         ; If initial rotation N/comp cont.
          BIT.B     #0x10, &P1IN                  ; If initial rotat. comp. test lmt swtch
          JC        TarFin                        ; If lmt switch is set; Target finished
Step8:    MOV.B     &P4OUT, temp                  ; Moves current P4 outputs into temp
          BIC.B     #0xF0, temp                   ; Clears out top nibble of temp P4
          BIS.B     #0x50, temp                   ; Sets the current step bits in temp
          MOV.B     R15, &P4OUT                   ; Moves the new value of P4 to P4out
          MOV.W     R6, R14                       ; Loads the Timer_B intial count (35ms)
          CALL      #TDelay                       ; Calls Time Delay Sub-routine
          BIT.B     #0x10, Status                 ; Tests for initial rotation complete
          JNC       Step9                         ; If initial rotation N/comp cont.
          BIT.B     #0x10, &P1IN                  ; If initial rotat. comp. test lmt swtch
          JC        TarFin                        ; If lmt switch is set; Target finished
Step9:    DEC.W     TarRot                        ; 8 primary steps complete dec. counter
          JNZ       Step1                         ; If counter not zero start 8 steps agn
          BIS.B     #0x10, Status                 ; If zero; set ini. rot. comp. stat. bit
          MOV.W     #0xE700, R6                   ; Load longer time delay for Timer_B Dly
          JMP       Step1                         ; Start 8 steps looking for lmt swtch
TarFin:   BIC.B     #0xF0, &P4OUT                 ; De-energizes motor coils
          BIS.B     #0x08, &P4OUT                 ; Disable DC-DC Converter & Limit Switch
          MOV.W     #0x01, Dly2OL                 ; Loads Outer Delay2 loop for 0.1s delay
          CALL      #TDelay2                      ; Calls Time Delay #2 routine
          BIC.W     #0x10, &TACCTL0               ; Clears Capture 0 interrupt enable
          BIC.W     #0x10, &TACCTL1               ; Clears Capture 1 interrupt enable
          BIC.B     #0x08, &P1IFG                 ; P1.3 IFG Cleared
          BIS.B     #0x01, Status                 ; Sets Task Complete Bit in Status
          BIC.B     #0x10, Status                 ; Clears initial rotation complete flag
          RET                                     ; Return

; Received Change Mode Command ----------------------------------------------
RcvMode:MOV.W     #0x02, cmd                      ; Selects 3rd byte of Receive Cmd
          BIT.B     #0x01, RcvCmd(R13)            ; Tests for Mode 1 or 3
          JC        Mode1                         ; If Mode 1 or 3 then jmp to Mode 1
          BIT.B     #0x02, RcvCmd(R13)            ; Tests for Mode 2 or 3
          JC        Mode2                         ; If Mode 2 or 3 then jmp to Mode 2
          CALL      #SndNAK                       ; Else, send a NAK
          JMP       FlshSet                       ; Jump to return
Mode1:    ; Normal Operation Mode 1, Light curtain on in front no rear
          BIT.B     #0x02, RcvCmd(R13)            ; Tests for Mode 3
          JC        Mode3                         ; If Mode 3 then jmp to Mode 3
```

```
         BIC.B    #0x06, Status             ; Clears current mode
         BIS.B    #0x02, Status             ; Else, Set Mode 1 Status Bit
         BIC.B    #0x08, &P1IE              ; P1.3 Interrupt disabled
         BIC.B    #0x08, &P1REN             ; Disable P1.3 Pull-up Resistor
         BIC.B    #0x08, &P1OUT             ; Force P1.3 Output low
         BIS.B    #0x08, &P1DIR             ; P1.3 output selected
         BIC.B    #0x08, &P1IFG             ; P1.3 IFG Cleared
         BIS.B    #0x04, &P2OUT             ; Sets P2.2, Front Light Curtain ON
         BIC.B    #0x08, &P2OUT             ; Clrs P2.3, Rear Light Curtain OFF
         CALL     #CapSet                   ; Call the Capture Setup routine
         BIS.W    #0x10, &TACCTL0           ; Capture 0 interrupt enable
         BIC.W    #0x10, &TACCTL1           ; Clears Capture 1 intterupt enable
         MOV.W    #0x0005, Dly2OL           ; Loads Outer Delay2 loop
         CALL     #TDelay2                  ; Calls Time Delay
         BIC.W    #0x01, &TACCTL0           ; Capture 0 interrupt flag clear
         BIC.W    #0x01, &TACCTL1           ; Capture 1 intterupt flag clear
         BIC.B    #0x08, &P1IFG             ; P1.3 IFG Cleared
         JMP      FlshSet                   ; Jump to return
Mode2:   ; Chronograph Mode 2, Both Light Curtains on front and rear
         BIC.B    #0x08, &P1IE              ; P1.3 Interrupt disabled
         BIC.B    #0x08, &P1REN             ; Disable P1.3 Pull-up Resistor
         BIC.B    #0x08, &P1OUT             ; Force P1.3 Output low
         BIS.B    #0x08, &P1DIR             ; P1.3 output selected
         BIC.B    #0x08, &P1IFG             ; P1.3 IFG Cleared
         BIT.B    #0x01, RcvCmd(R13)        ; Tests for Mode 3
         JC       Mode3                     ; If Mode 3 then jmp to Mode 3
         BIC.B    #0x06, Status             ; Clears current mode
         BIS.B    #0x04, Status             ; Else, Set Mode 2 Status Bit
         BIS.B    #0x0C, &P2OUT             ; Sets P2.2&3, Both Light Curtains ON
         CALL     #CapSet                   ; Call the Capture Setup routine
         BIS.W    #0x10, &TACCTL0           ; Capture 0 interrupt enable
         BIS.W    #0x10, &TACCTL1           ; Capture 1 intterupt enable
         MOV.W    #0x0005, Dly2OL           ; Loads Outer Delay2 loop
         CALL     #TDelay2                  ; Calls Time Delay
         BIC.W    #0x01, &TACCTL0           ; Capture 0 interrupt flag clear
         BIC.W    #0x01, &TACCTL1           ; Capture 1 intterupt flag clear
         BIC.B    #0x08, &P1IFG             ; P1.3 IFG Cleared
         JMP      FlshSet                   ; Jump to return
Mode3:   ; Demo Mode 3, Both Light Curtains off, Laser-gun input active
         BIC.B    #0x06, Status             ; Clears current mode
         BIS.B    #0x06, Status             ; Set Mode 3 Status Bits
         BIC.B    #0x08, &P1DIR             ; Selects active P1 Inputs
         BIS.B    #0x08, &P1REN             ; Activates Pull-?(-P1OUT-) Resistor
         BIS.B    #0x08, &P1OUT             ; Selected pins inputs are pulled high
         BIS.B    #0x08, &P1IES             ; P1.3 Select High to Low transition
         BIC.B    #0x08, &P1IFG             ; P1.3 IFG Cleared
         BIC.W    #0x10, &TACCTL0           ; Clears Capture 0 interrupt enable
         BIC.W    #0x10, &TACCTL1           ; Clears Capture 1 intterupt enable
         BIC.B    #0x08, &P1IFG             ; P1.3 IFG Cleared
         BIC.B    #0x0C, &P2OUT             ; Clears P2.2&3, Both Light Curtains OFF
         BIS.B    #0x80, &P3OUT             ; Sets P3.8, Laser Gun laser OFF
         BIS.B    #0x08, &P1IE              ; P1.3 Interrupt enabled
FlshSet: ; Camera Flash set-up
         INC.W    cmd                       ; Increments Cmd offset
         BIT.B    #0x01, RcvCmd(R13)        ; Tests first bit in 4th byte of cmd
         JC       FlshOn                    ; If bit set enable camera flash
         BIC.B    #0x08, Status             ; If not set disable camera flash
         JMP      ModDun                    ; Jump to Mode Change Done
FlshOn:  BIS.B    #0x08, Status             ; If bit set enable camera flash
ModDun:  BIS.B    #0x01, Status             ; Sets Task Complete Bit in Status
         MOV.W    #0x0005, Dly2OL           ; Loads Outer Delay2 loop
         CALL     #TDelay2                  ; Calls Time Delay
         BIC.W    #0x01, &TACCTL0           ; Capture 0 interrupt flag clear
         BIC.W    #0x01, &TACCTL1           ; Capture 1 intterupt flag clear
         RET                                ; Return from sub-routine

; Received Command Containing Shot Time ---------------------------------------
RcvTme:  BIS.B    #0x01, Status             ; Sets Task Complete Bit in Status
         RET                                ; Return

; Send ACK Command ------------------------------------------------------------
SndACK:  CLR.W    cmd                       ; Clears the Command Array offset
```

```
         MOV.B    #CmdPrfx, SndCmd(R13)    ; SndCmd[0] = CmdPrfx
         INC.W    cmd                      ; Increments Command Array offset
         MOV.B    #CmdACK, SndCmd(R13)     ; SndCmd[1] = CmdACK
         MOV.B    RcvCmd(R13), temp        ; Temp = last Rcv'd Cmd
         INC.W    cmd                      ; Increments Command Array offset
         MOV.B    R15, SndCmd(R13)         ; SndCmd[2] = Last Rcv'd Cmd
         MOV.B    RcvCmd(R13), temp        ; Moves RCmd[2] to SCmd[3]
         INC.W    cmd                      ; Increments Command Array offset
         MOV.B    R15, SndCmd(R13)         ; Moves 0x00 to 4th byte of SndCmd
         CALL     #TxCmd                   ; Calls Transmit Command sub-routine
         RET                               ; Return from sub-routine

; Send NAK Command ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
SndNAK:  CLR.W    cmd                      ; Clears the Command Array offset
         MOV.B    #CmdPrfx, SndCmd(R13)    ; SndCmd[0] = CmdPrfx
         MOV.B    RcvCmd(R13), j           ; j = last Rcv'd Prfx
         INC.W    cmd                      ; Increments Command Array offset
         MOV.B    #CmdNAK, SndCmd(R13)     ; SndCmd[1] = CmdNAK
         MOV.B    RcvCmd(R13), k           ; k = last Rcv'd Cmd
         INC.W    cmd                      ; Increments Command Array offset
         MOV.B    R5, SndCmd(R13)          ; SndCmd[2] = Last Rcv'd Prfx
         INC.W    cmd                      ; Increments Command Array offset
         MOV.B    R6, SndCmd(R13)          ; SndCmd[3] = Last Rcv'd Cmd
         CALL     #TxCmd                   ; Calls Transmit Command sub-routine
         RET                               ; Return from sub-routine

; Time Delay ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
TDelay:  BIS.W    #TBCLR, &TBCTL           ; Clears prev. timer_b value & settings
         BIC.W    #TBIFG, &TBCTL           ; Clears Interrupt Flag
         MOV.W    #TBSSEL_2+ID_3, &TBCTL   ; Timer_B: SMCLK/8 = 500kHz, Stopped
         MOV.W    R14, &TBR                ; Loads Timer_B IniDly (R14)
         BIS.W    #MC_2, &TBCTL            ; Starts Timer B for cont. mode
TWait:   BIT.W    #TBIFG, &TBCTL           ; Waits for Timer_B Overflow
         JNC      TWait                    ; If no overflow jump to TWait
         BIS.W    #TBCLR, &TBCTL           ; Clears prev. timer_b value & settings
         BIC.W    #TBIFG, &TBCTL           ; Clears Interrupt Flag
         BIC.W    #MC_2, &TBCTL            ; Stops Timer B for cont. mode
         RET                               ; Return

; Time Delay #2 No Timers approx. 100ms per outer loop ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
TDelay2:
TWait2:  MOV.W    #0xBE00, Dly2IL          ; Timer #2 inner loop creates 100ms dly
TWait3:  DEC.W    Dly2IL                   ; Decrements Inner loop by one
TWait4:  CMP.W    #0x00, Dly2IL            ; Compares Inner loop to zero
         JNZ      TWait3                   ; If not zero loop again
         DEC.W    Dly2OL                   ; If zero, then decrement outer loop
         CMP.W    #0x00, Dly2OL            ; Compares Outer loop to zero
         JNZ      TWait2                   ; If not zero do both loops again
         RET                               ; If zero delay is finished, Return

; Transmit Command ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
TxCmd:
Wt4DTR:  BIT.B    #0x40, &P1IN             ; Tests P1.6 for Bluetooth Connection
         JNC      Wt4DTR                   ; If Low, Not Connected, Wait
         CLR.W    cmd                      ; Clears the Command Array offset
TxFAgn:  BIT.B    #UCA0TXIFG, &IFG2        ; Checks the interrupt flag in Tx Status
         JNC      TxFAgn                   ; If Tx not finished jump to TxFAgn
         MOV.B    SndCmd(R13), &UCA0TXBUF  ; Send Current Cmd Byte to TxBuff
         INC.W    cmd                      ; Increment Command Array offset
         CMP.W    #0x04, cmd               ; Compare R13 to 4
         JNZ      TxFAgn                   ; If not equal, then Transmit Next Byte
         RET                               ; Return from sub-routine

;~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
;~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Interrupt Routines ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
;~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

; Serial Receive Interrupt Routine ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
SerISR:  PUSH.W   Dly2OL                   ; Pushes old Delay 2 outer loop counter
         BIC.B    #GIE, SR                 ; All enabled Interrupts disabled
         BIT.B    #0x40, &P1IN             ; Tests P1.6 for Bluetooth Connection
         JNC      NoWay                    ; If Low, Not Connected to Micro
```

```
          CLR.W    cmd                        ; Clears Command Array offset
NxtByt:   MOV.W    #0x7FFF, 1                 ; Sets Timeout loop counter
RxFAgn:   BIT.B    #UCA0RXIFG, &IFG2          ; Rx Interrupt Flag AND IFG2
          JC       RdBuf                      ; Rx Receive Int Flag is set Read Buffer
          DEC.W    1                          ; Else Decrement Timeout loop counter
          JNZ      RxFAgn                     ; If no timeout Check Rx Flag again
          JZ       S_TOut                     ; If timeout jump to Serial Timeout
RdBuf:    MOV.B    &UCA0RXBUF, RcvCmd(R13)    ; Moves RxBuff into RcvCmd[R13]
          INC.W    cmd                        ; Increments Array offset
          CMP.W    #0x04, cmd                 ; Subtracts Array offset from 4
          JNZ      NxtByt                     ; If not zero -> not finished, Next Byte
          CALL     #ChkCmd                    ; If zero, then Check the Cmd
          POP.W    Dly2OL                     ; Pops old Delay 2 Outer loop counter
          BIC.W    #CCIFG, &TACCTL0           ; Clear the Capture/Compare Int Flag
          BIC.W    #CCIFG, &TACCTL1           ; Clear the Capture/Compare Int Flag
          BIS.B    #GIE, SR                   ; All enabled Interrupts enabled
          RETI                                ; Return from interrupt
S_TOut:   CALL     #SndNAK                    ; Serial Receive Timeout, Send NAK
NoWay:    POP.W    Dly2OL                     ; Pops old Delay 2 Outer loop counter
          BIC.W    #CCIFG, &TACCTL0           ; Clear the Capture/Compare Int Flag
          BIC.W    #CCIFG, &TACCTL1           ; Clear the Capture/Compare Int Flag
          BIS.B    #GIE,SR                    ; All enabled Interrupts enabled
          RETI                                ; Return from interrupt

; Laser Gun Input Interrupt Routine ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
LGunISR:BIT.B      #0x40, &P3OUT              ; Test for servo laser state
          JC       ShtNG                      ; If servo laser OFF, then shot too late
          MOV.W    Dly2OL, shot2              ; Else, move shot timer value to shot2
          JMP      LGun                       ; Then jump to LGun
ShtNG:    MOV.W    #0x0000, shot2             ; If shot NG, move zero to shot timer
          JMP      LGun                       ; Then jump to LGun
LGun:     PUSH.W   Dly2OL                     ; Pushes old Delay 2 Outer loop counter
          BIC.B    #GIE, SR                   ; All enabled Interrupts disabled
          CALL     #DmoSht                    ; Calls DemoShot routine
          BIC.B    #0x08, &P1IFG              ; P1.3 Interrupt Flag Cleared
          POP.W    Dly2OL                     ; Pops old Delay 2 Outer loop counter
          BIS.B    #GIE, SR                   ; All enabled Interrupts enabled
          RETI                                ; Return from interrupt

; Timer_A_0 Capture Routine ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
T_Capt0:MOV.W      &TACCR0, shot1             ; Move Timer_A Capt Val. (P1.1) to shot1
          PUSH.W   Dly2OL                     ; Pushes old Delay 2 Outer loop counter
          BIC.B    #GIE, SR                   ; All enabled Interrupts disabled
          BIT.B    #0x40, &P1IN               ; Tests P1.6 for Bluetooth Connection
          JNC      NoShot1                    ; If Low, BT connected to Camera
          BIC.W    #CCIFG, &TACCTL0           ; Clear the Capture/Compare Int Flag
          MOV.B    Status, temp               ; Moves Status to temp
          AND.B    #0x6, temp                 ; Masks to keep mode bits
          CMP.B    #0x2, temp                 ; Checks for mode 1, No Chrono
          JZ       Mode1a                     ; Jump to Mode 1a
          POP.W    Dly2OL                     ; Pops old Delay 2 Outer loop counter
          BIC.W    #CCIFG, &TACCTL0           ; Clear the Capture/Compare Int Flag
          BIC.W    #CCIFG, &TACCTL1           ; Clear the Capture/Compare Int Flag
          BIS.B    #GIE, SR                   ; All enabled Interrupts enabled
          RETI                                ; Return from interrupt
Mode1a:   CLR.W    cmd                        ; Clears the Command Array offset
          MOV.B    #CmdPrfx, SndCmd(R13)      ; SndCmd[0] = CmdPrfx
          INC.W    cmd                        ; Increments Command Array offset
          MOV.B    #CmdShot, SndCmd(R13)      ; SndCmd[1] = CmdShot
          INC.W    cmd                        ; Increments Command Array offset
          MOV.W    #00, SndCmd(R13)           ; SndCmd[2,3] = Shot Time (dt)
          CALL     #TxCmd                     ; Calls Transmit Command sub-routine
          CALL     #CamWait                   ; Calls the Camera and Flash routine
NoShot1:POP.W      Dly2OL                     ; Pops old Delay 2 Outer loop counter
          BIC.W    #CCIFG, &TACCTL0           ; Clear the Capture/Compare Int Flag
          BIC.W    #CCIFG, &TACCTL1           ; Clear the Capture/Compare Int Flag
          BIS.B    #GIE, SR                   ; All enabled Interrupts enabled
          RETI                                ; Returns from interrupt

; Timer_A_1 Capture Routine ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
T_Capt1:MOV.W      &TACCR1, shot2             ; Move Timer_A Capt Value (P1.2) shot2
          PUSH.W   Dly2OL                     ; Pushes old Delay 2 Outer loop counter
```

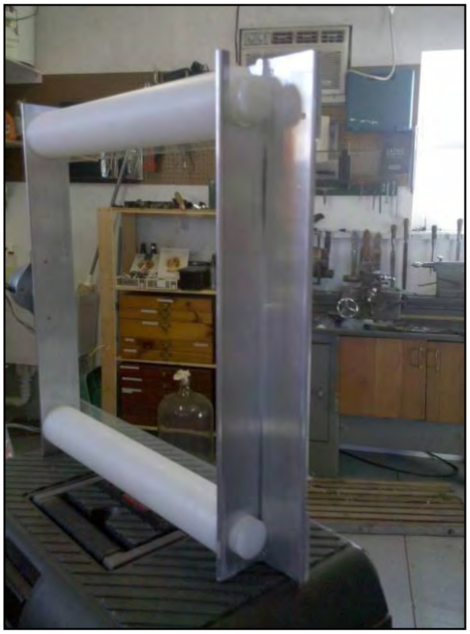
```
          BIC.B    #GIE, SR                  ; All enabled Interrupts disabled
          BIT.B    #0x40, &P1IN              ; Tests P1.6 for Bluetooth Connection
          JNC      NoShot2                   ; If Low, BT connected to Camera
          BIC.W    #CCIFG, &TACCTL1          ; Clear the Capture/Compare Int Flag
          MOV.W    R11, dt                   ; Moves shot1 (R10) to R12 for sub
          SUB.W    R10, dt                   ; Subtracts shot2 (R11) from shot1
          SWPB     dt                        ; Swaps bytes of Shot Time for SndCmd
          CLR.W    cmd                       ; Clears the Command Array offset
          MOV.B    #CmdPrfx, SndCmd(R13)     ; SndCmd[0] = CmdPrfx
          INC.W    cmd                       ; Increments Command Array offset
          MOV.B    #CmdShot, SndCmd(R13)     ; SndCmd[1] = CmdShot
          INC.W    cmd                       ; Increments Command Array offset
          MOV.W    R12, SndCmd(R13)          ; SndCmd[2,3] = Shot Time (dt)
          CALL     #TxCmd                    ; Calls Transmit Command sub-routine
          CALL     #CamWait                  ; Calls the Camera and Flash routine
NoShot2:  POP.W    Dly2OL                    ; Pops old Delay 2 Outer loop counter
          BIC.W    #CCIFG, &TACCTL0          ; Clear the Capture/Compare Int Flag
          BIC.W    #CCIFG, &TACCTL1          ; Clear the Capture/Compare Int Flag
          BIS.B    #GIE, SR                  ; All enabled Interrupts enabled
          RETI                               ; Return from interrupt

; Interrupt Vectors ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
          COMMON   INTVEC                    ; Interrupt Vectors
          ORG      PORT1_VECTOR              ; Port 1 input interrupt vector address
          DW       LGunISR                   ; Sets Laser Gun interrupt vctor LGunISR
          ORG      USCIAB0RX_VECTOR          ; UART Rx vector address
          DW       SerISR                    ; Sets UART Rx vector to 'SerISR' label
          ORG      TIMERA1_VECTOR            ; Timer_A_1 interrupt vector address
          DW       T_Capt1                   ; Sets Timer Capture 1 vector to T_Capt1
          ORG      TIMERA0_VECTOR            ; Timer_A_0 interrupt vector address
          DW       T_Capt0                   ; Sets Timer Capture 0 vector to T_Capt0
          ORG      RESET_VECTOR              ; MSP430 RESET Vector
          DW       RESET                     ; Sets reset vector to 'RESET' label

END                                          ; That's all Folks
```

APPENDIX B

Photographs

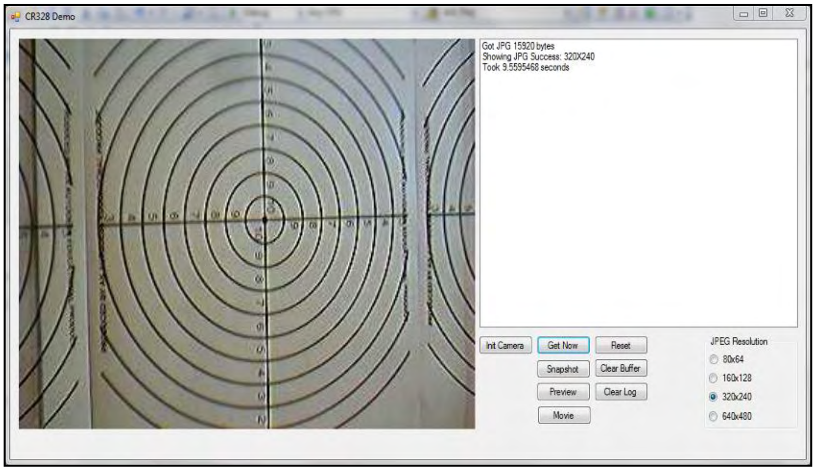


Target Advance System

C328R Serial Camera

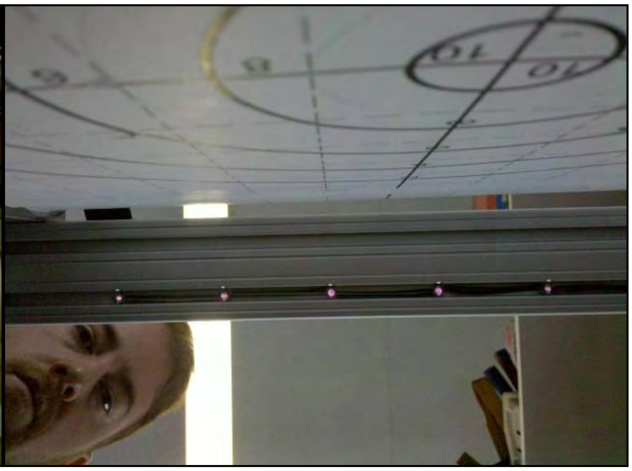
Camera and Flash Permanently Mounted

Platform's Main Structure



First Image of a Target from the Serial Camera

Platform's Main Structure and Target



Infrared Emitters Turned On

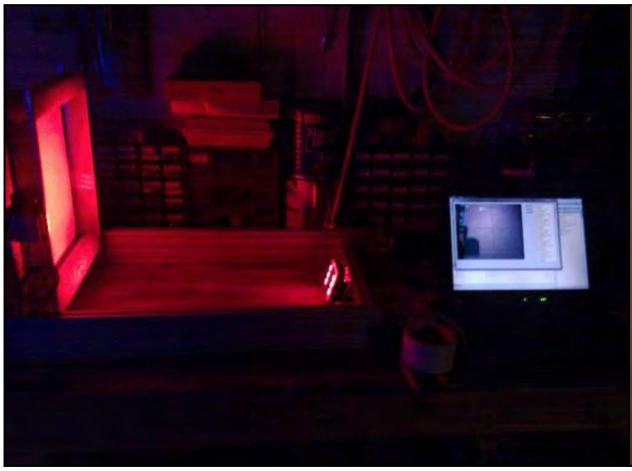
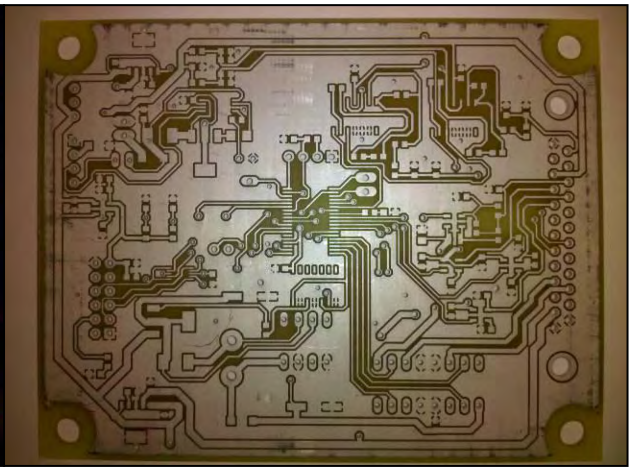
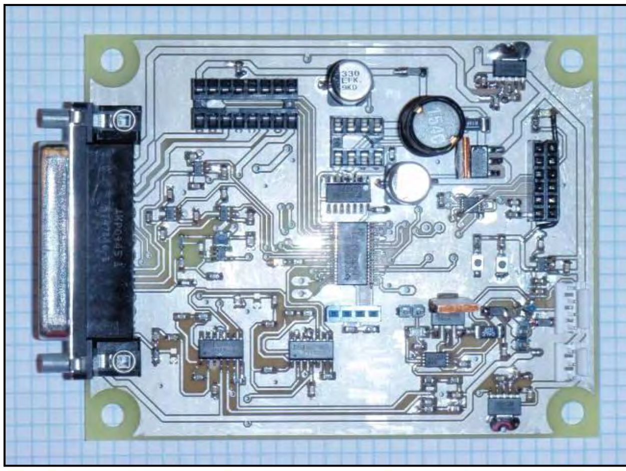


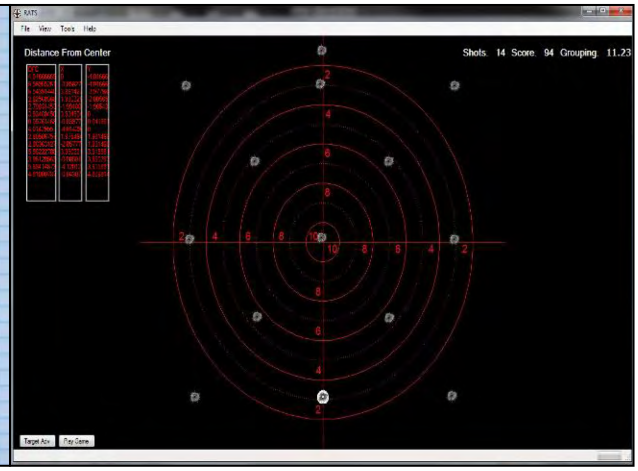
Image Testing with Flash



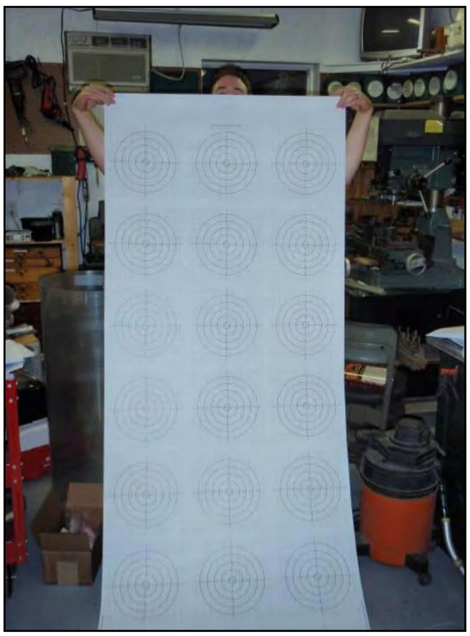
Printed Circuit Board without Components



Printed Circuit Board with Components

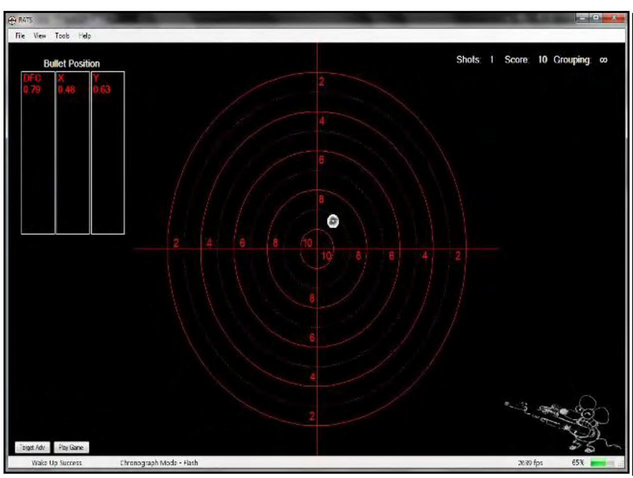


Screenshot of Calibration Analysis



Targets Printed from Plotter

Finished Main Control Box

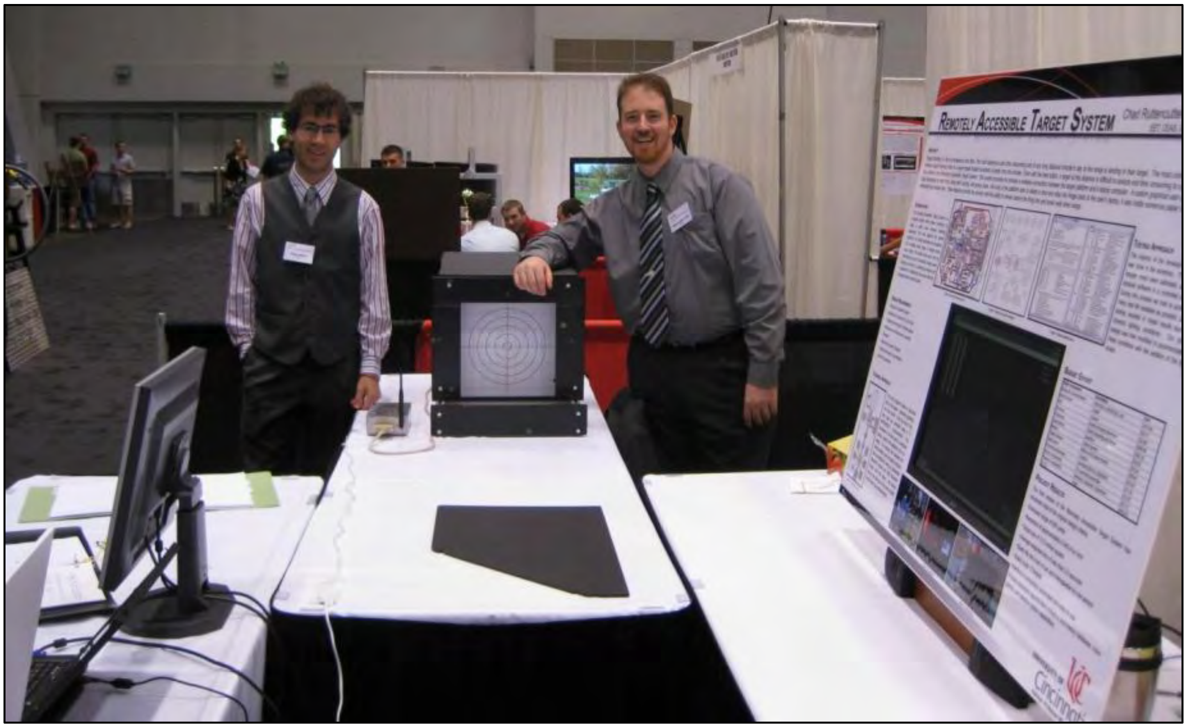


Screenshot from Actual Bullet (17hmr)

RATS ‘About’ form with Logo

Remotely Accessible Target System at its Completion

Tech Expo 2010, Remotely Accessible Target System



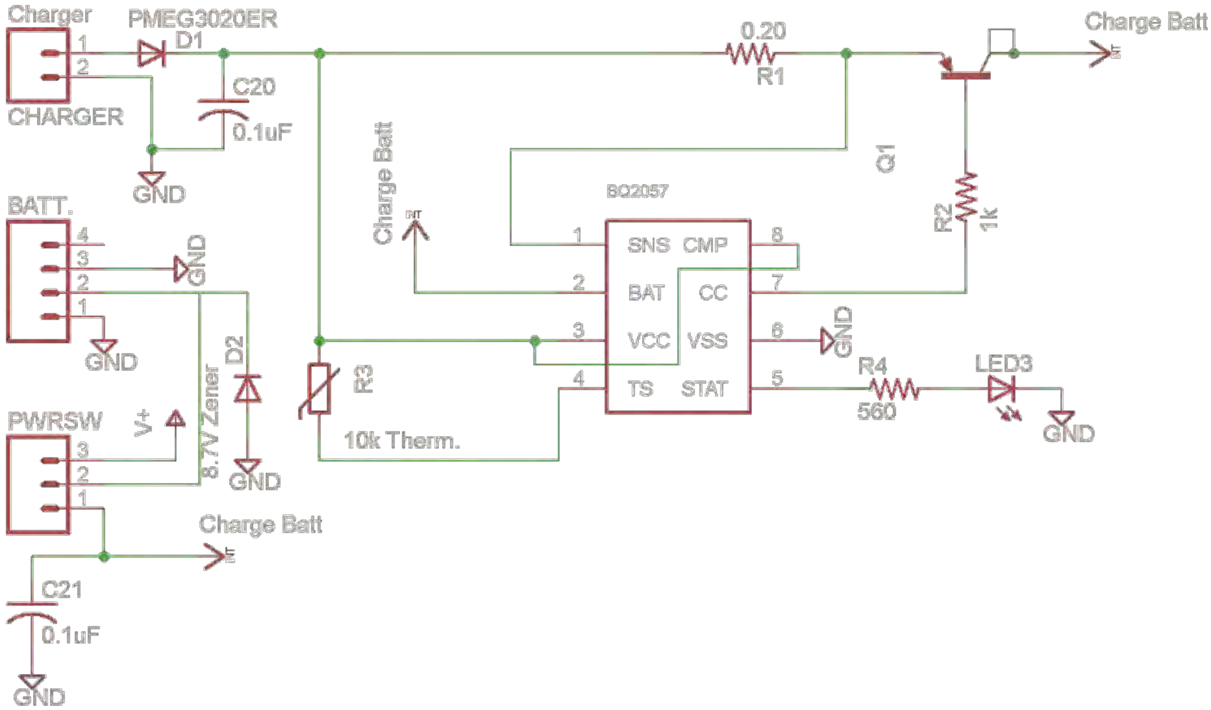
Tech Expo 2010, Remotely Accessible Target System

Tech Expo Awards: Best EET, Best IEEE, Harris Communications Award, and Most Innovative Concept.
Left to Right: Peter Simon, Professor Michael Haas, Chad Ruttencutter

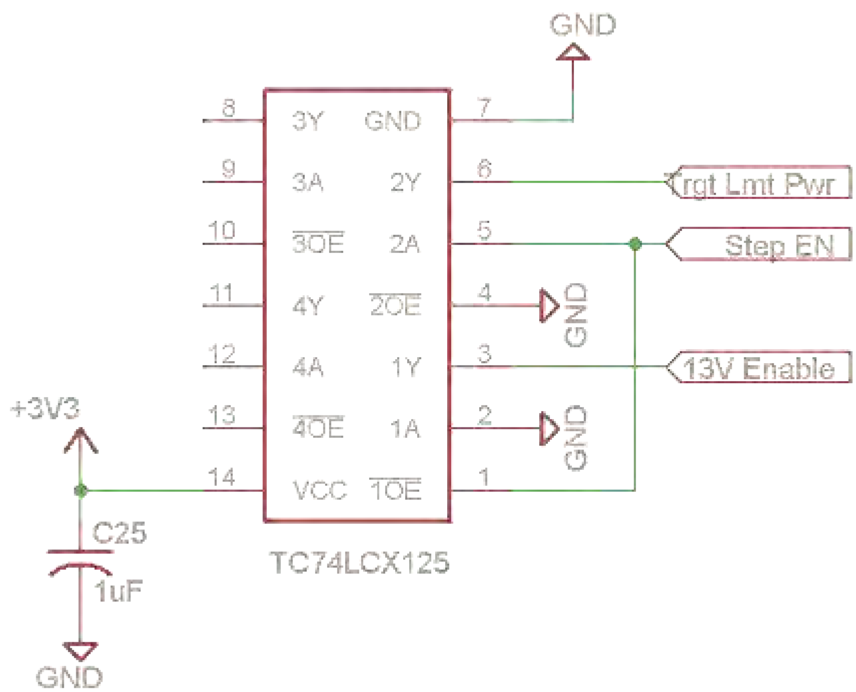
APPENDIX C

Schematics

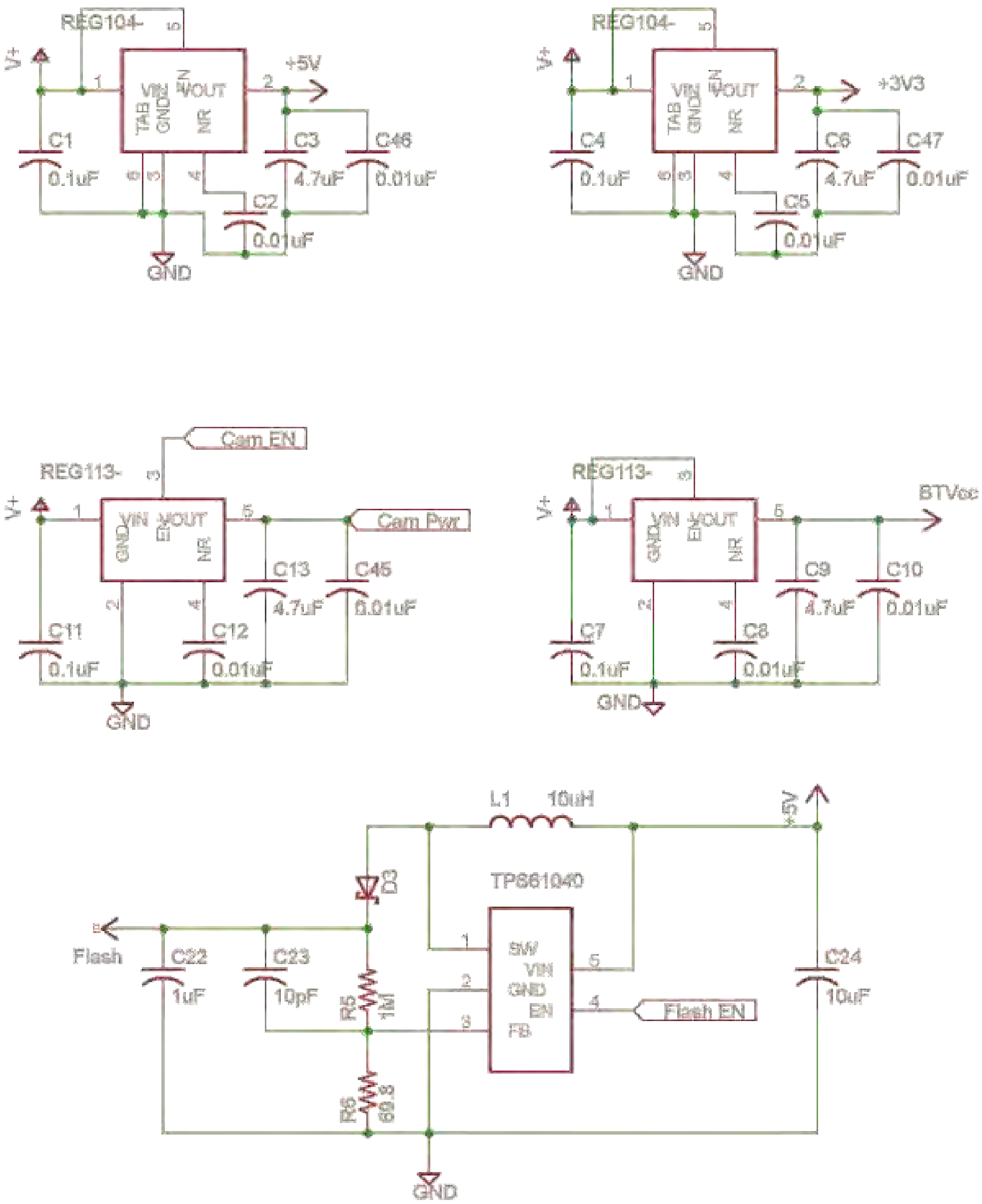
Battery and Charging Circuit



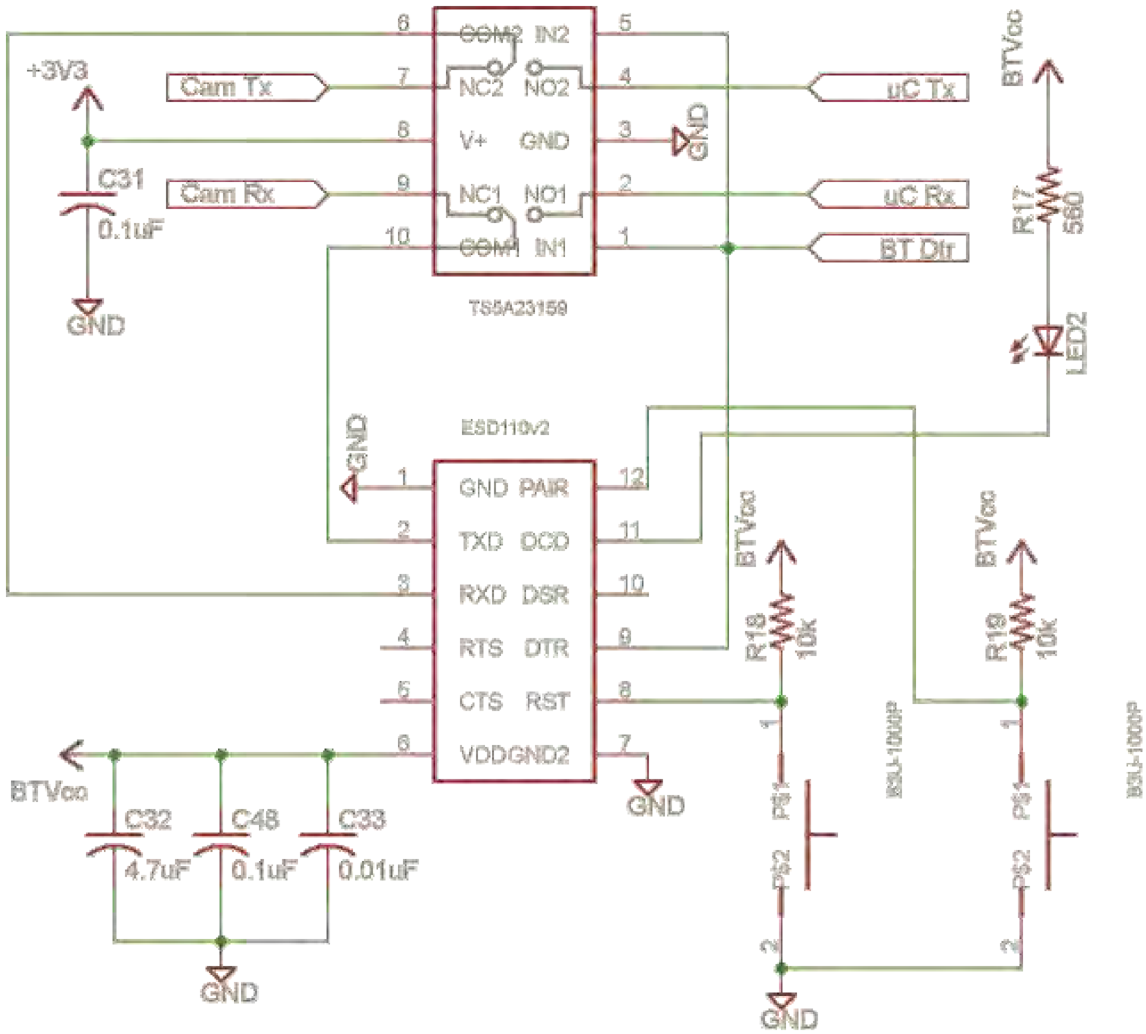
Output Buffer Circuit



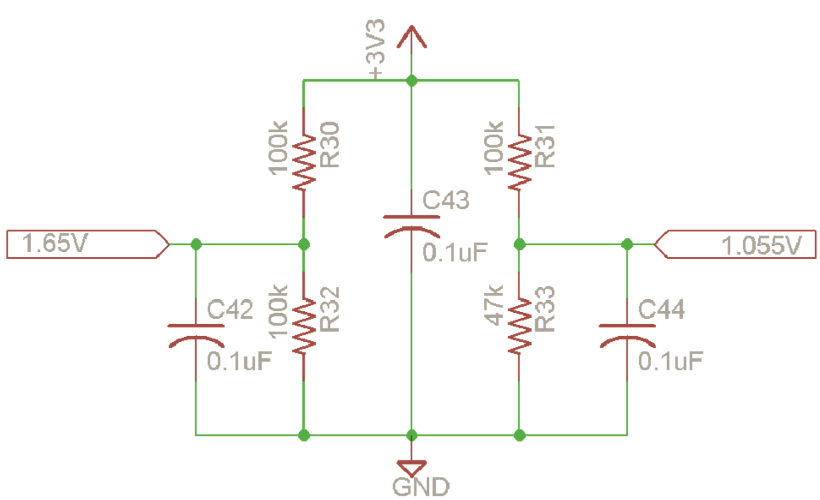
Power Regulation Circuits



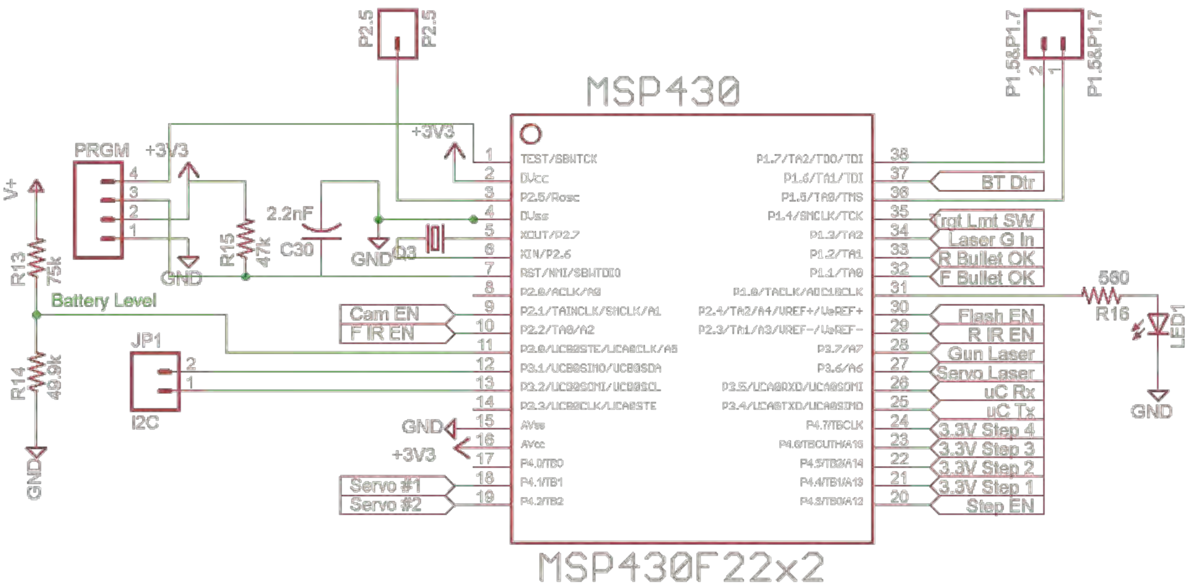
Communication Circuit



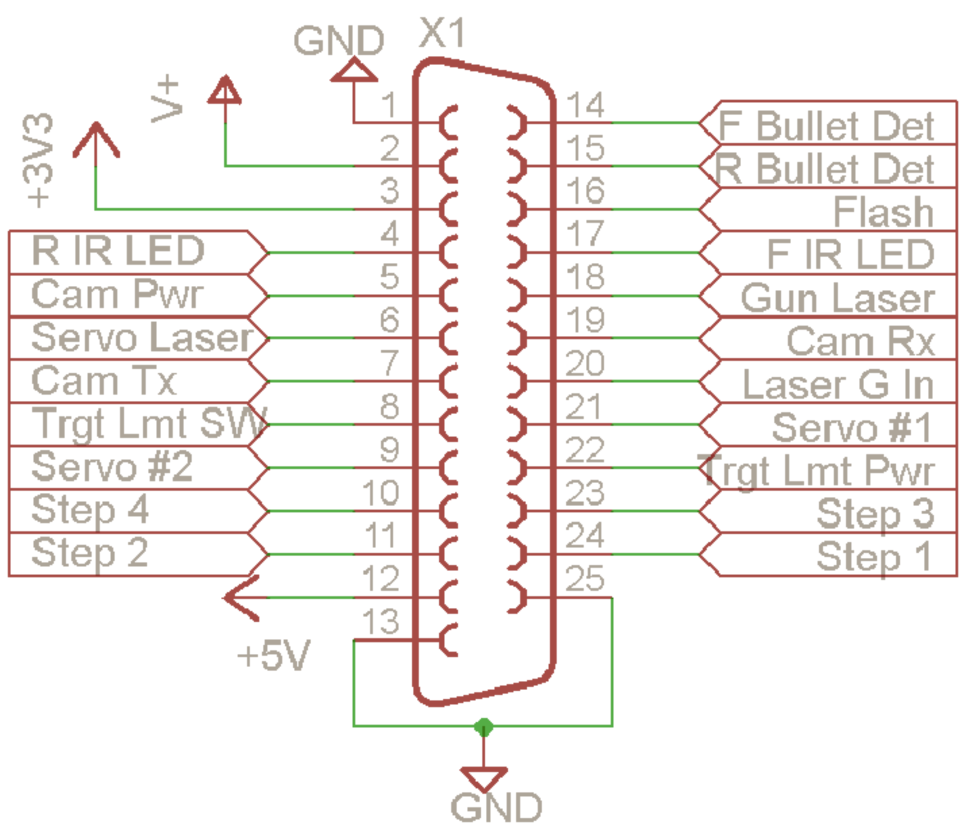
Amplifier Voltage Divider Reference Circuit



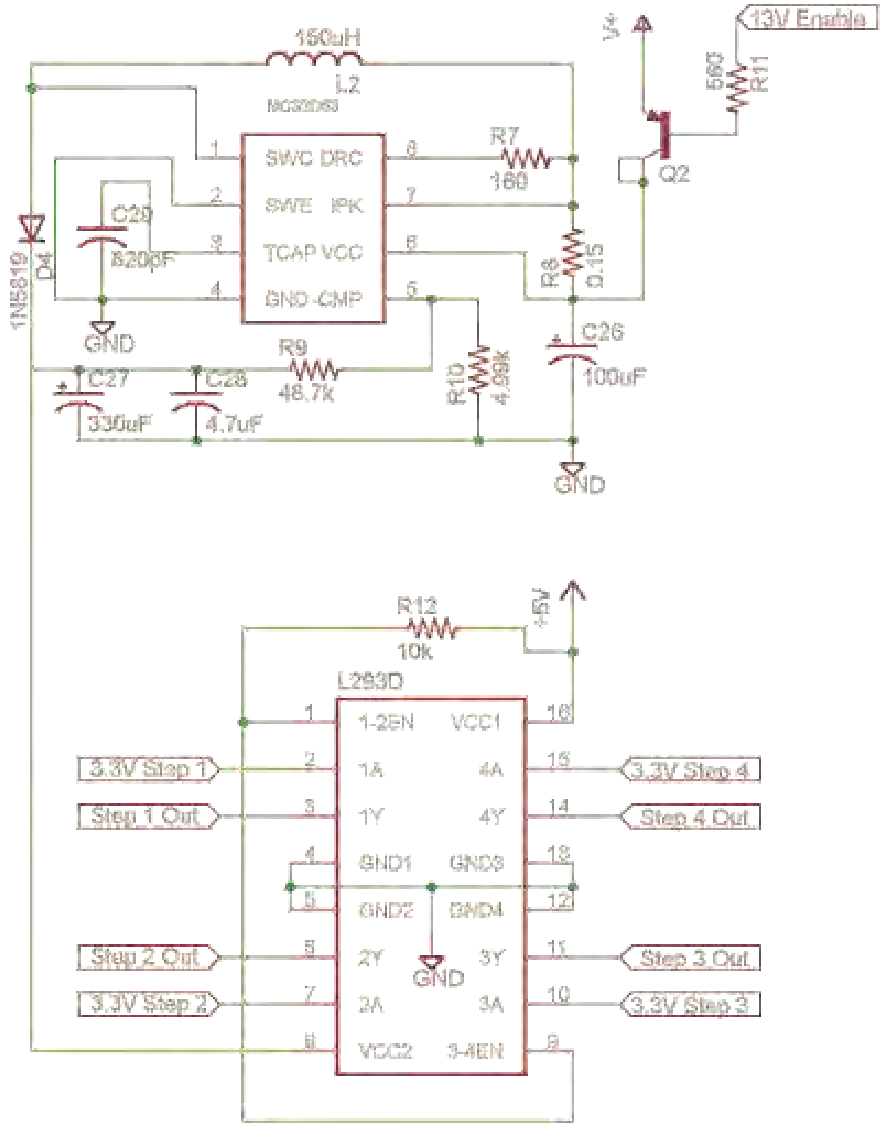
MSP430f2272 Microcontroller Circuit



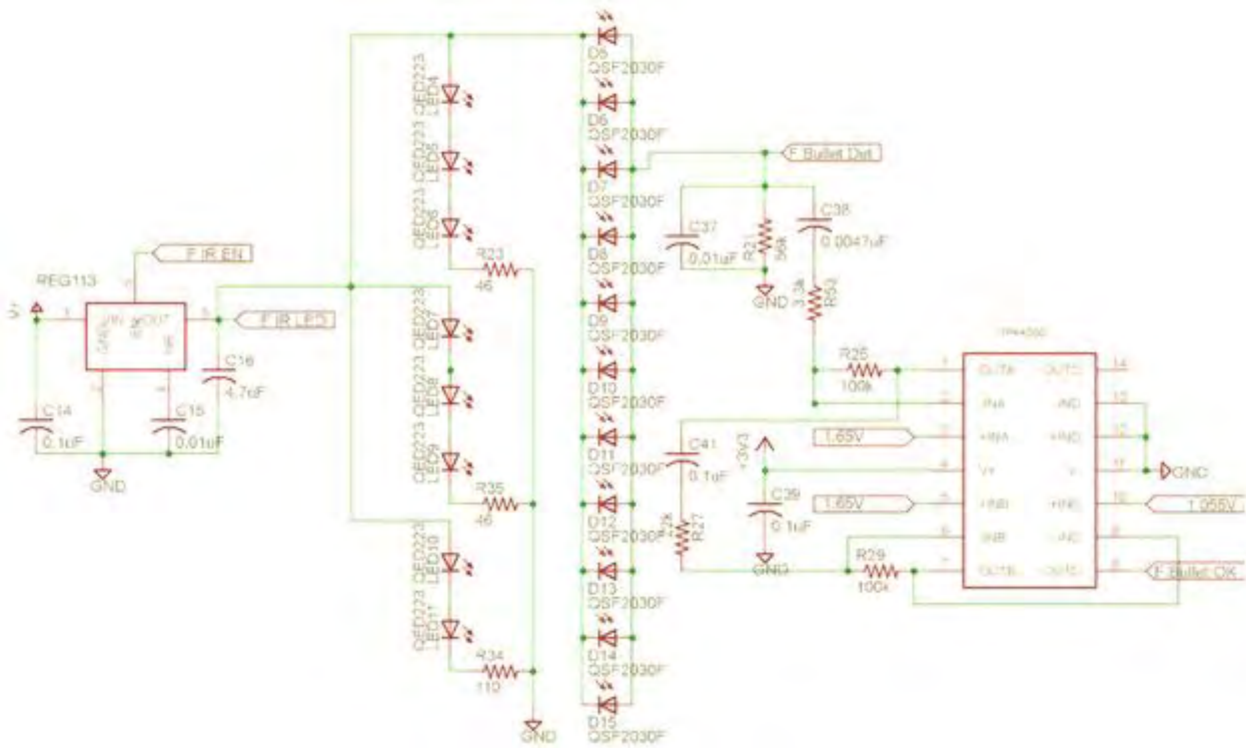
Control Box to Platform Connector Circuit



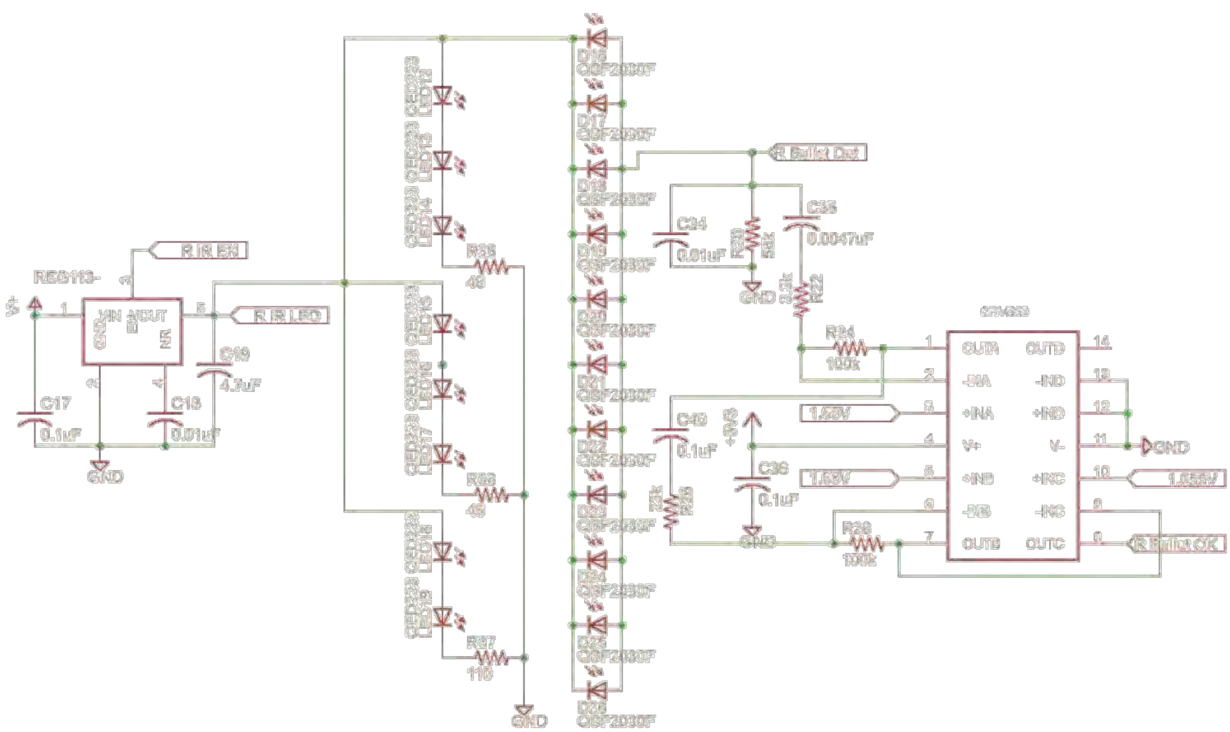
Stepper Motor Circuit



Front Bullet Detection Circuit



Rear Bullet Detection Circuit



APPENDIX D

Datasheets

MSP430f2272

MSP430x22x2, MSP430x22x4
MIXED SIGNAL MICROCONTROLLER

SLAS504C – JULY 2006 – REVISED MAY 2009

- Low Supply Voltage Range: 1.8 V to 3.6 V
- Ultralow Power Consumption
  - Active Mode: 270 µA at 1 MHz, 2.2 V
  - Standby Mode: 0.7 µA
  - Off Mode (RAM Retention): 0.1 µA
- Ultrafast Wake-Up From Standby Mode in Less Than 1 µs
- 16-Bit RISC Architecture, 62.5-ns Instruction Cycle Time
- Basic Clock Module Configurations:
  - Internal Frequencies up to 16 MHz With Four Calibrated Frequencies to ±1%
  - Internal Very-Low-Power Low-Frequency Oscillator
  - 32-kHz Crystal
  - High-Frequency Crystal up to 16 MHz
  - Resonator
  - External Digital Clock Source
  - External Resistor
- 16-Bit Timer_A With Three Capture/Compare Registers
- 16-Bit Timer_B With Three Capture/Compare Registers
- Universal Serial Communication Interface
  - Enhanced UART Supporting Auto-Baudrate Detection (LIN)
  - IrDA Encoder and Decoder
  - Synchronous SPI
  - I^2C™
- 10-Bit 200-ksps Analog-to-Digital (A/D) Converter With Internal Reference, Sample-and-Hold, Autoscan, and Data Transfer Controller
- Two Configurable Operational Amplifiers (MSP430x22x4 Only)
- Brownout Detector
- Serial Onboard Programming, No External Programming Voltage Needed Programmable Code Protection by Security Fuse
- Bootstrap Loader
- On Chip Emulation Module
- Family Members Include:
  MSP430F2232: 8KB + 256B Flash Memory 512B RAM
  MSP430F2252: 16KB + 256B Flash Memory 512B RAM
  MSP430F2272: 32KB + 256B Flash Memory 1KB RAM
  MSP430F2234: 8KB + 256B Flash Memory 512B RAM
  MSP430F2254: 16KB + 256B Flash Memory 512B RAM
  MSP430F2274: 32KB + 256B Flash Memory 1KB RAM
- Available in a 38-Pin Thin Shrink Small-Outline Package (TSSOP) (DA) and 40-Pin QFN Package (RHA) (See Available Options)
- For Complete Module Descriptions, See the *MSP430x2xx Family User's Guide*, Literature Number SLAU144

description

The Texas Instruments MSP430 family of ultralow-power microcontrollers consist of several devices featuring different sets of peripherals targeted for various applications. The architecture, combined with five low-power modes is optimized to achieve extended battery life in portable measurement applications. The device features a powerful 16-bit RISC CPU, 16-bit registers, and constant generators that contribute to maximum code efficiency. The digitally controlled oscillator (DCO) allows wake-up from low-power modes to active mode in less than 1 µs.

This integrated circuit can be damaged by ESD. Texas Instruments recommends that all integrated circuits be handled with appropriate precautions. Failure to observe proper handling and installation procedures can cause damage. ESD damage can range from subtle performance degradation to complete device failure. Precision integrated circuits may be more susceptible to damage because very small parametric changes could cause the device not to meet its published specifications. These devices have limited built-in ESD protection.

Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

ESD110v2

Parani-ESD100V2/110V2

User Guide

Version 2.0.0

2009-11-19

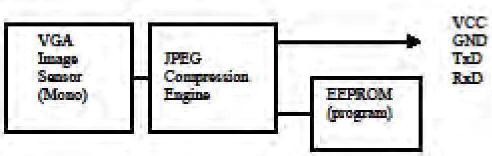
C328R

C328-7221
Mono Camera Module
W/ UART Interface

General Description

The C328-7221 is VGA camera module which is specially designed for serial communication and working under low light condition. It performs as a JPEG compressed still camera and can be attached to a wireless or PDA host. Users can send out a snapshot command from the host in order to capture a full resolution single-frame still picture. The picture is then compressed by the JPEG engine and transferred to the host thru serial port.

Block Diagram



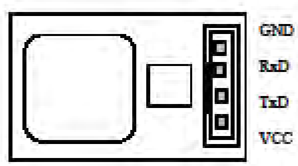
Features

- Small in size, 20x28mm
- Monochrome image sensor used
- 8-bit, 256-level grey level
- Sensitive to IR, good for night vision
- VGA resolution, down sample to QVGA
- 3.3V operation
- Low power consumption 105mA
- User friendly commands for control
- UART interface of up to 115.2Kbps
- Auto detect baud rate and make connection to the host
- Power saving mode
- Various lens options

Pin Description

Pin	Description
VCC	Power 3.3VDC
TxD	Data Transmit (3.3V)
RxD	Data Receive (3.3V)
GND	Power Ground

Connector specification: 2mm pitch, 4pin single row
Reference part no: Suyin 190600
Mating connector: Suyin 140600



Bottom View

Command Summary

Detail Command control, please refer to the user's manual. Please note the command set is same as C328-7640, only some parameters are different.

1.	Initial	To configure the image size
2.	Get Picture	Get Picture type
3.	Snapshot	Set snap shot image type
4.	Set Package Size	Set the package size to transmit data from module to Host
5.	Set Baudrate	Change the baud rate
6.	Reset	Reset the whole system or reset the state machine
7.	Power Off	To enter sleep mode
8.	Data	Set the data type and length for transmitting data to host
9.	SYNC	Sync signal to connect between host and module
10.	ACK	Command to indicate the communication success
11.	NAK	Command to indicate the communication fail with error code

Rev 1.0

REG104

Burr-Brown Products from Texas Instruments

REG104

SBVS025G – SEPTEMBER 2001 – REVISED SEPTEMBER 2005

DMOS 1A Low-Dropout Regulator

FEATURES

- **NEW DMOS TOPOLOGY:**
  Ultra Low Dropout Voltage:
  230mV typ at 1A and 3.3V Output
  Output Capacitor NOT Required for Stability
- **FAST TRANSIENT RESPONSE**
- **VERY LOW NOISE: 33µV_{RMS}**
- **HIGH ACCURACY: ±2% max**
- **HIGH EFFICIENCY:**
  I_{GND} **= 1.7mA at** I_{OUT} **= 1A**
  Not Enabled: I_{GND} **= 0.5µA**
- **2.5V, 2.7V, 3.0V, 3.3V, 5.0V AND ADJUSTABLE OUTPUT VERSIONS**
- **THERMAL PROTECTION**
- **SMALL SURFACE-MOUNT PACKAGES: SOT223-5, DDPAK-5**

APPLICATIONS

- **PORTABLE COMMUNICATION DEVICES**
- **BATTERY-POWERED EQUIPMENT**
- **MODEMS**
- **BAR-CODE SCANNERS**
- **BACKUP POWER SUPPLIES**

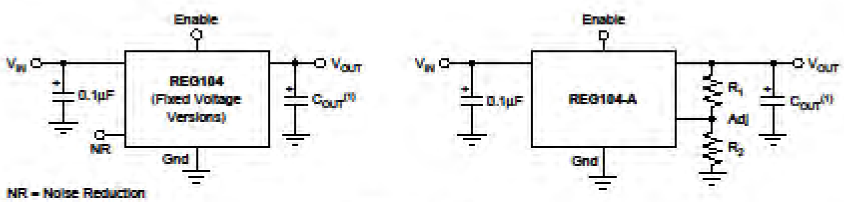
DESCRIPTION

The REG104 is a family of low-noise, low-dropout linear regulators with low ground pin current. Its new DMOS topology provides significant improvement over previous designs, including low dropout voltage (only 230mV typ at full load), and better transient performance. In addition, no output capacitor is required for stability, unlike conventional low dropout regulators that are difficult to compensate and require expensive low ESR capacitors greater than 1µF.

Typical ground pin current is only 1.7mA (at I_{OUT} = 1A) and drops to 0.5µA in *not enabled* mode. Unlike regulators with PNP pass devices, quiescent current remains relatively constant over load variations and under dropout conditions.

The REG104 has very low output noise (typically 33µV_{RMS} for V_{OUT} = 3.3V with C_{NR} = 0.01µF), making it ideal for use in portable communications equipment. On-chip trimming results in high output voltage accuracy. Accuracy is maintained over temperature, line, and load variations. Key parameters are tested over the specified temperature range (–40°C to +85°C).

The REG104 is well protected—internal circuitry provides a current limit which protects the load from damage. Thermal protection circuitry keeps the chip from being damaged by excessive temperature. The REG104 is available in the DDPAK-5 and the SOT223-5.



NOTE: (1) Optional.

Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

Texas Instruments
www.ti.com

REG113

Burr-Brown Products from Texas Instruments

REG113

SBVS031D – MARCH 2001 – REVISED SEPTEMBER 2005

DMOS 400mA Low-Dropout Regulator

FEATURES

- CAP-FREE DMOS TOPOLOGY:
  Ultra Low Dropout Voltage:
  250mV typ at 400mA
  Output Capacitor *not* Required for Stability
- UP TO 500mA PEAK, TYPICAL
- FAST TRANSIENT RESPONSE
- VERY LOW NOISE: 28µVrms
- HIGH ACCURACY: ±1.5% max
- HIGH EFFICIENCY:
  I_{GND} = 850µA at I_{OUT} = 400mA
  Not Enabled: I_{GND} = 0.01µA
- 2.5V, 2.85V, 3.0V, 3.3V, AND 5.0V OUTPUT VERSIONS
- OTHER OUTPUT VOLTAGES AVAILABLE UPON REQUEST
- FOLDBACK CURRENT LIMIT
- THERMAL PROTECTION
- SMALL SURFACE-MOUNT PACKAGES:
  SOT23-5 and MSOP-8

APPLICATIONS

- PORTABLE COMMUNICATION DEVICES
- BATTERY-POWERED EQUIPMENT
- PERSONAL DIGITAL ASSISTANTS
- MODEMS
- BAR-CODE SCANNERS
- BACKUP POWER SUPPLIES

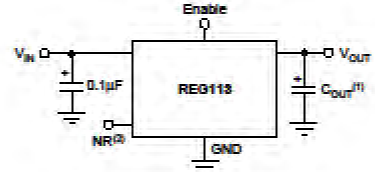
DESCRIPTION

The REG113 is a family of low-noise, low-dropout linear regulators with low ground pin current. Its new DMOS topology provides significant improvement over previous designs, including low-dropout voltage (only 250mV typ at full load), and better transient performance. In addition, no output capacitor is required for stability, unlike conventional low-dropout regulators that are difficult to compensate and require expensive low ESR capacitors greater than 1µF.

Typical ground pin current is only 850µA (at I_{OUT} = 400mA) and drops to 10nA when not enabled. Unlike regulators with PNP pass devices, quiescent current remains relatively constant over load variations and under dropout conditions.

The REG113 has very low output noise (typically 28µVrms for V_{OUT} = 3.3V with C_{NR} = 0.01µF), making it ideal for use in portable communications equipment. Accuracy is maintained over temperature, line, and load variations. Key parameters are tested over the specified temperature range (–40°C to +85°C).

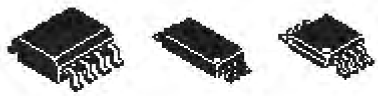
The REG113 is well protected—internal circuitry provides a current limit which protects the load from damage, furthermore, thermal protection circuitry keeps the chip from being damaged by excessive temperature. The REG113 is available in SOT23-5 and MSOP-8 packages.



NOTES: (1) Optional. (2) NR = Noise Reduction.

BQ2057t

bq2057, bq2057C
bq2057T, bq2057W



SLUS025F – MAY 2001 – REVISED JULY 2002

ADVANCED LINEAR CHARGE MANAGEMENT IC FOR SINGLE- AND TWO-CELL LITHIUM-ION AND LITHIUM-POLYMER

FEATURES

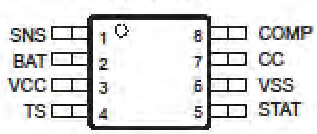
- **Ideal for Single (4.1 V or 4.2 V) and Dual-Cell (8.2 V or 8.4 V) Li-Ion or Li-Pol Packs**
- **Requires Small Number of External Components**
- **0.3 V Dropout Voltage for Minimizing Heat Dissipation**
- **Better Than ±1% Voltage Regulation Accuracy With Preset Voltages**
- **AutoComp™ Dynamic Compensation of Battery Pack's Internal Impedance to Reduce Charge Time**
- **Optional Cell-Temperature Monitoring Before and During Charge**
- **Integrated Voltage and Current Regulation With Programmable Charge-Current and High- or Low-Side Current Sensing**
- **Integrated Cell Conditioning for Reviving Deeply Discharged Cells and Minimizing Heat Dissipation During Initial Stage Of Charge**
- **Charge Status Output for Single or Dual Led or Host Processor Interface**
- **Automatic Battery-Recharge Feature**
- **Charge Termination by Minimum Current**
- **Automatic Low-Power Sleep Mode When V_{CC} Is Removed**
- **EVMs Available for Quick Evaluation**
- **Packaging: 8-Pin SOIC, 8-Pin TSSOP, 8-Pin MSOP**

DESCRIPTION

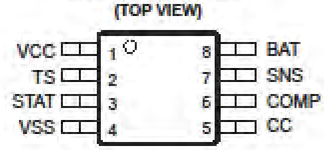
The BENCHMARQ bq2057 series advanced Lithium-Ion (Li-Ion) and Lithium-Polymer (Li-Pol) linear charge-management ICs are designed for cost-sensitive and compact portable electronics. They combine high-accuracy current and voltage regulation, battery conditioning, temperature monitoring, charge termination, charge-status indication, and AutoComp charge-rate compensation in a single 8-pin IC. MSOP, TSSOP, and SOIC package options are offered to fit a wide range of end applications.

The bq2057 continuously measures battery temperature using an external thermistor. For safety, the bq2057 inhibits charge until the battery temperature is within user-defined thresholds. The bq2057 then charges the battery in three phases: conditioning, constant current, and constant voltage. If the battery voltage is below the low-voltage threshold, $V_{(min)}$, the bq2057 precharges using a low current to condition the battery. The conditioning charge rate is approximately 10% of the regulation current. The conditioning current also minimizes heat dissipation in the external pass-element during the initial stage of the charge. After conditioning, the bq2057 applies a constant current to the battery. An external sense-resistor sets the current. The sense-resistor can be on either the high or low side of the battery without additional components. The constant-current phase continues until the battery reaches the charge-regulation voltage.

bq2057xSN or bq2057xTS
SOIC (SN) or TSSOP (TS) PACKAGE
(TOP VIEW)



bq2057xDGK
MSOP (DGK) PACKAGE
(TOP VIEW)

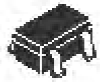


Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

AutoComp is a trademark of Texas Instruments.

TPS61040

TEXAS INSTRUMENTS
www.ti.com



TPS61040
TPS61041

SLVS413C–OCTOBER 2002–REVISED AUGUST 2007

LOW-POWER DC/DC BOOST CONVERTER IN SOT-23 AND SON PACKAGES

FEATURES

- **1.8-V to 6-V Input Voltage Range**
- **Adjustable Output Voltage Range up to 28 V**
- **400-mA (TPS61040) and 250-mA (TPS61041) Internal Switch Current**
- **Up to 1-MHz Switching Frequency**
- **28-μA Typical No-Load Quiescent Current**
- **1-μA Typical Shutdown Current**
- **Internal Soft Start**
- **Available in SOT23-5 and 2 × 2 × 0.8-mm SON Packages**

APPLICATIONS

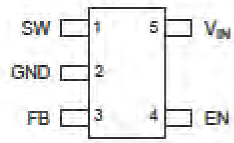
- **LCD Bias Supply**
- **White-LED Supply for LCD Backlights**
- **Digital Still Camera**
- **PDAs, Organizers, and Handheld PCs**
- **Cellular Phones**
- **Internet Audio Player**
- **Standard 3.3-V/5-V to 12-V Conversion**

DESCRIPTION

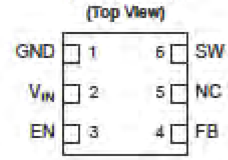
The TPS61040/41 is a high-frequency boost converter dedicated for small to medium LCD bias supply and white LED backlight supplies. The device is ideal to generate output voltages up to 28 V from a dual cell NiMH/NiCd or a single cell Li-Ion battery. The part can also be used to generate standard 3.3-V/5-V to 12-V power conversions.

The TPS61040/41 operates with a switching frequency up to 1 MHz. This allows the use of small external components using ceramic as well as tantalum output capacitors. Together with the thin SON package, the TPS61040/41 gives a very small overall solution size. The TPS61040 has an internal 400 mA switch current limit, while the TPS61041 has a 250-mA switch current limit, offering lower output voltage ripple and allows the use of a smaller form factor inductor for lower power applications. The low quiescent current (typically 28 μA) together with an optimized control scheme, allows device operation at very high efficiencies over the entire load current range.

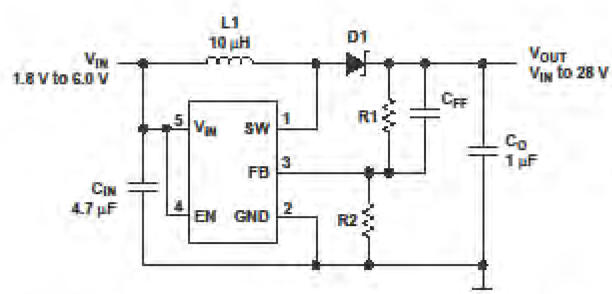
DBV PACKAGE
(Top View)



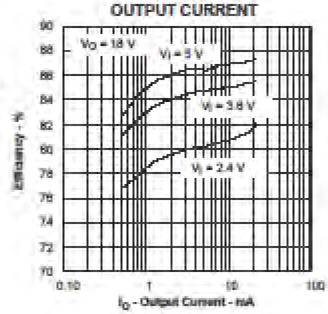
DRV PACKAGE
(Top View)



TYPICAL APPLICATION



EFFICIENCY
vs
OUTPUT CURRENT



Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

TS5A23159

TEXAS INSTRUMENTS

TS5A23159

www.ti.com SCDS201E–AUGUST 2005–REVISED JULY 2009

1-Ω DUAL SPDT ANALOG SWITCH 5-V/3.3-V 2-CHANNEL 2:1 MULTIPLEXER/DEMULTIPLEXER

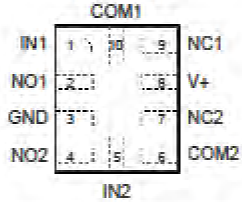
FEATURES

- Isolation in Power-Down Mode, $V_+ = 0$
- Specified Break-Before-Make Switching
- Low ON-State Resistance (1 Ω)
- Control Inputs Are 5.5-V Tolerant
- Low Charge Injection
- Excellent ON-State Resistance Matching
- Low Total Harmonic Distortion (THD)
- 1.65-V to 5.5-V Single-Supply Operation
- Latch-Up Performance Exceeds 100 mA Per JESD 78, Class II
- ESD Performance Tested Per JESD 22
  - 2000-V Human-Body Model (A114-B, Class II)
  - 1000-V Charged-Device Model (C101)

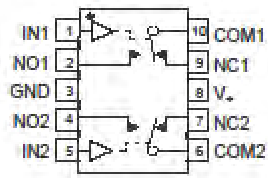
APPLICATIONS

- Cell Phones
- PDAs
- Portable Instrumentation
- Audio and Video Signal Routing
- Low-Voltage Data-Acquisition Systems
- Communication Circuits
- Modems
- Hard Drives
- Computer Peripherals
- Wireless Terminals and Peripherals

RSE PACKAGE (TOP VIEW)



DGS PACKAGE (TOP VIEW)



DESCRIPTION/ORDERING INFORMATION

The TS5A23159 is a dual single-pole double-throw (SPDT) analog switch that is designed to operate from 1.65 V to 5.5 V. The device offers low ON-state resistance and excellent ON-state resistance matching with the break-before-make feature, to prevent signal distortion during the transferring of a signal from one channel to another. The device has an excellent total harmonic distortion (THD) performance and consumes very low power. These features make this device suitable for portable audio applications.

ORDERING INFORMATION(1)

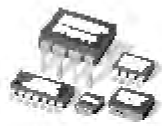
T_A	PACKAGE(2)		ORDERABLE PART NUMBER	TOP-SIDE MARKING
–40°C to 85°C	VSSOP – DGS (MSOP)	Tape and reel	TS5A23159DGSR	JER or JEO
	QFN – RSE	Tape and reel	TS5A23159RSER	JEO

(1) For the most current package and ordering information, see the Package Option Addendum at the end of this document, or see the TI web site at www.ti.com.
(2) Package drawings, thermal data, and symbolization are available at www.ti.com/packaging.

Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

OPA4350

Burr-Brown Products from Texas Instruments



OPA350 OPA2350 OPA4350

SBOS099C – SEPTEMBER 2000 – REVISED JANUARY 2005

High-Speed, Single-Supply, Rail-to-Rail OPERATIONAL AMPLIFIERS *MicroAmplifier*™ Series

FEATURES

- RAIL-TO-RAIL INPUT
- RAIL-TO-RAIL OUTPUT (within 10mV)
- WIDE BANDWIDTH: 38MHz
- HIGH SLEW RATE: 22V/µs
- LOW NOISE: 5nV/√Hz
- LOW THD+NOISE: 0.0006%
- UNITY-GAIN STABLE
- *Micro*SIZE PACKAGES
- SINGLE, DUAL, AND QUAD

APPLICATIONS

- CELL PHONE PA CONTROL LOOPS
- DRIVING A/D CONVERTERS
- VIDEO PROCESSING
- DATA ACQUISITION
- PROCESS CONTROL
- AUDIO PROCESSING
- COMMUNICATIONS
- ACTIVE FILTERS
- TEST EQUIPMENT

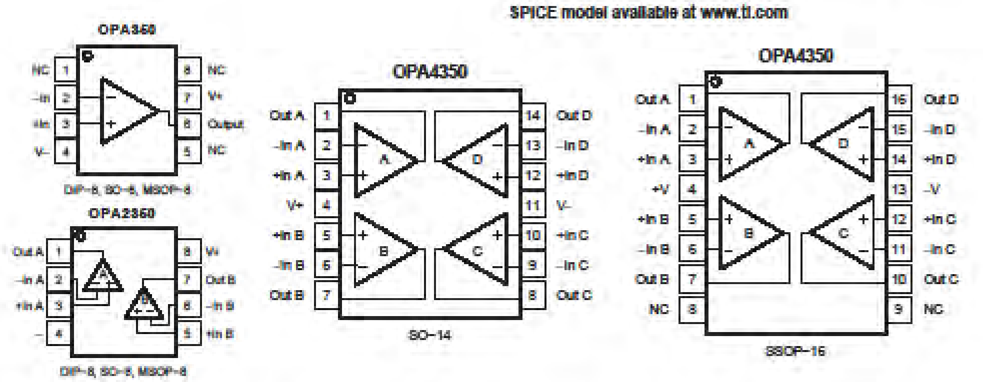
DESCRIPTION

The OPA350 series rail-to-rail CMOS operational amplifiers are optimized for low voltage, single-supply operation. Rail-to-rail input/output, low noise (5nV/√Hz), and high speed operation (38MHz, 22V/µs) make them ideal for driving sampling Analog-to-Digital (A/D) converters. They are also well suited for cell phone PA control loops and video processing (75Ω drive capability) as well as audio and general purpose applications. Single, dual, and quad versions have identical specifications for maximum design flexibility.

The OPA350 series operates on a single supply as low as 2.5V with an input common-mode voltage range that extends 300mV below ground and 300mV above the positive supply. Output voltage swing is to within 10mV of the supply rails with a 10kΩ load. Dual and quad designs feature completely independent circuitry for lowest crosstalk and freedom from interaction.

The single (OPA350) and dual (OPA2350) come in the miniature MSOP-8 surface mount, SO-8 surface mount, and DIP-8 packages. The quad (OPA4350) packages are the space-saving SSOP-16 surface mount and SO-14 surface mount. All are specified from –40°C to +85°C and operate from –55°C to +150°C.

SPICE model available at www.ti.com



Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

All trademarks are the property of their respective owners.

MC33063A

TEXAS INSTRUMENTS

MC33063A
MC34063A

www.ti.com SLLS636L – DECEMBER 2004 – REVISED DECEMBER 2009

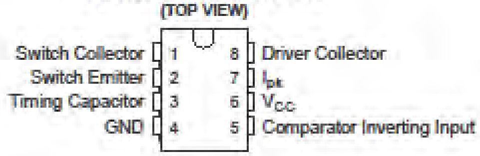
1.5-A PEAK BOOST/BUCK/INVERTING SWITCHING REGULATORS

Check for Samples: MC33063A MC34063A

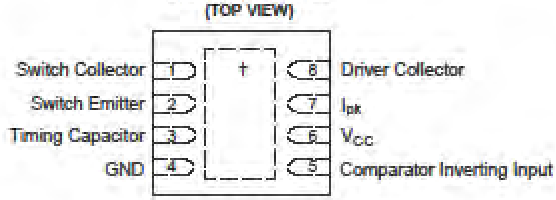
FEATURES

- **Wide Input Voltage Range: 3 V to 40 V**
- **High Output Switch Current: Up to 1.5 A**
- **Adjustable Output Voltage**
- **Oscillator Frequency Up to 100 kHz**
- **Precision Internal Reference: 2%**
- **Short-Circuit Current Limiting**
- **Low Standby Current**

D (SOIC) OR P (PDIP) PACKAGE
(TOP VIEW)



DRJ (QFN) PACKAGE
(TOP VIEW)



† The exposed thermal pad is electrically bonded internally to pin 4 (GND).

DESCRIPTION/ORDERING INFORMATION

The MC33063A and MC34063A are easy-to-use ICs containing all the primary circuitry needed for building simple dc-dc converters. These devices primarily consist of an internal temperature-compensated reference, a comparator, an oscillator, a PWM controller with active current limiting, a driver, and a high-current output switch. Thus, the devices require minimal external components to build converters in the boost, buck, and inverting topologies.

The MC33063A is characterized for operation from –40°C to 85°C, while the MC34063A is characterized for operation from 0°C to 70°C.

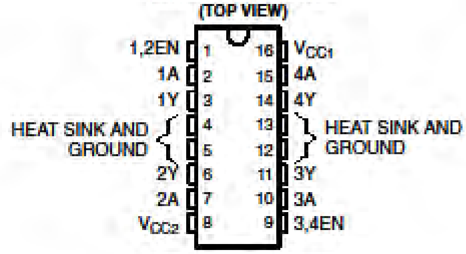
L293D

L293, L293D
QUADRUPLE HALF-H DRIVERS

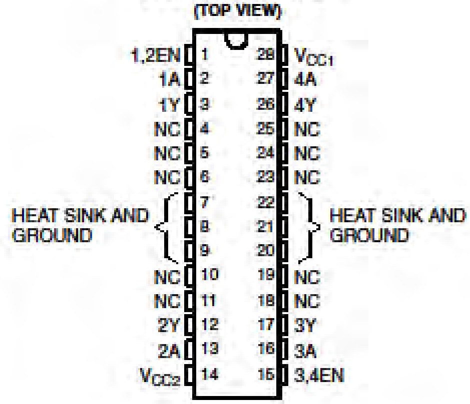
SLRS008C – SEPTEMBER 1986 – REVISED NOVEMBER 2004

- **Featuring Unitrode L293 and L293D Products Now From Texas Instruments**
- **Wide Supply-Voltage Range: 4.5 V to 36 V**
- **Separate Input-Logic Supply**
- **Internal ESD Protection**
- **Thermal Shutdown**
- **High-Noise-Immunity Inputs**
- **Functionally Similar to SGS L293 and SGS L293D**
- **Output Current 1 A Per Channel (600 mA for L293D)**
- **Peak Output Current 2 A Per Channel (1.2 A for L293D)**
- **Output Clamp Diodes for Inductive Transient Suppression (L293D)**

L293 . . . N OR NE PACKAGE
L293D . . . NE PACKAGE
(TOP VIEW)



L293 . . . DWP PACKAGE
(TOP VIEW)



description/ordering information

The L293 and L293D are quadruple high-current half-H drivers. The L293 is designed to provide bidirectional drive currents of up to 1 A at voltages from 4.5 V to 36 V. The L293D is designed to provide bidirectional drive currents of up to 600-mA at voltages from 4.5 V to 36 V. Both devices are designed to drive inductive loads such as relays, solenoids, dc and bipolar stepping motors, as well as other high-current/high-voltage loads in positive-supply applications.

All inputs are TTL compatible. Each output is a complete totem-pole drive circuit, with a Darlington transistor sink and a pseudo-Darlington source. Drivers are enabled in pairs, with drivers 1 and 2 enabled by 1,2EN and drivers 3 and 4 enabled by 3,4EN. When an enable input is high, the associated drivers are enabled, and their outputs are active and in phase with their inputs. When the enable input is low, those drivers are disabled, and their outputs are off and in the high-impedance state. With the proper data inputs, each pair of drivers forms a full-H (or bridge) reversible drive suitable for solenoid or motor applications.

ORDERING INFORMATION

T_A	PACKAGE†		ORDERABLE PART NUMBER	TOP-SIDE MARKING
0°C to 70°C	HSOP (DWP)	Tube of 20	L293DWP	L293DWP
	PDIP (N)	Tube of 25	L293N	L293N
	PDIP (NE)	Tube of 25	L293NE	L293NE
		Tube of 25	L293DNE	L293DNE

† Package drawings, standard packing quantities, thermal data, symbolization, and PCB design guidelines are available at www.ti.com/sc/package.

Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

TC74LCX125

TOSHIBA TC74LCX125F/FN/FT/FK

TOSHIBA CMOS Digital Integrated Circuit Silicon Monolithic

TC74LCX125F,TC74LCX125FN,TC74LCX125FT,TC74LCX125FK

Low-Voltage Quad Bus Buffer with 5-V Tolerant Inputs and Outputs

Note: xxxFN (JEDEC SOP) is not available in Japan.

The TC74LCX125 is a high-performance CMOS quad bus buffers. Designed for use in 3.3-V systems, it achieves high-speed operation while maintaining the CMOS low power dissipation.

The device is designed for low-voltage (3.3 V) V_{CC} applications, but it could be used to interface to 5-V supply environment for inputs.

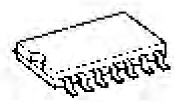
This device requires the 3-state control input $\overline{OE}$ to be set high to place the output into the high impedance state.

All inputs are equipped with protection circuits against static discharge.

Features

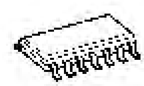
- Low-voltage operation: V_{CC} = 1.65 to 3.6 V
- High-speed operation: t_{pd} = 6.0 ns (max) (V_{CC} = 3.0 to 3.6 V)
- Ouput current: $|I_{OH}|/I_{OL}$ = 24 mA (min) (V_{CC} = 3.0 V)
- Latch-up performance: > ±500 mA
- Available in JEDEC SOP, JEITA SOP, TSSOP and VSSOP (US)
- Power-down protection is provided on all inputs and outputs
- Pin and function compatible with the 74 series (74AC/VHC/HC/F/ALS/LS etc.) 125 type

TC74LCX125F



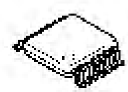
SOP14-P-300-1.27A

TC74LCX125FN



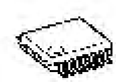
SOL14-P-150-1.27

TC74LCX125FT



TSSOP14-P-0044-0.65A

TC74LCX125FK



VSSOP14-P-0030-0.50

Weight

SOP14-P-300-1.27A	: 0.18 g (typ.)
SOL14-P-150-1.27	: 0.12 g (typ.)
TSSOP14-P-0044-0.65A	: 0.06 g (typ.)
VSSOP14-P-0030-0.50	: 0.02 g (typ.)

Note: The Electrical Characteristics of V_{CC}=1.8±0.15V is only applicable for products which manufactured from January 2009 onward.

FZT790A

FAIRCHILD SEMICONDUCTOR®

FZT790A

FZT790A

PNP Low Saturation Transistor

- These devices are designed with high current gain and low saturation voltage with collector currents up to 3A continuous.

SOT-223

1. Base 2.4. Collector 3. Emitter

Absolute Maximum Ratings * T_C=25°C unless otherwise noted

Symbol	Parameter	Value	Units
V_{CEO}	Collector-Emitter Voltage	-40	V
V_{CBO}	Collector-Base Voltage	-50	V
V_{EBO}	Emitter-Base Voltage	-5	V
I_C	Collector Current - Continuous	-3	A
T_J, T_{STG}	Operating and Storage Junction Temperature Range	- 55 ~ +150	°C

* These ratings are limiting values above which the serviceability of any semiconductor device may be impaired.

NOTES:
1. These ratings are based on a maximum junction temperature of 150degrees C.
2. These are steady state limits. The factory should be consulted on applications involving pulsed or low duty cycle operations.

Electrical Characteristics T_A=25°C unless otherwise noted

Symbol	Parameter	Test Conditions	Min.	Max.	Units
Off Characteristics					
BV_{CEO}	Collector-Emitter Breakdown Voltage	I_C = -10mA, I_B = 0	-40		V
BV_{CBO}	Collector-Emitter Breakdown Voltage	I_C = -100μA, I_E = 0	-50		V
BV_{EBO}	Emitter-Base Breakdown Voltage	I_E = -100μA, I_C = 0	-5.0		V
I_{CBO}	Collector Cut-off Current	V_{CB} = -30V, I_E = 0 V_{CB} = -30V, I_E = 0, T_A = 100°C		-100 -10	nA μA
I_{EBO}	Emitter Cut-off Current	V_{EB} = -4V, I_C = 0		-100	nA
On Characteristics *					
h_{FE}	DC Current Gain	V_{CE} = -2.0V, I_C = -10mA V_{CE} = -2.0V, I_C = -500mA V_{CE} = -2.0V, I_C = -1.0A V_{CE} = -2.0V, I_C = -2.0A	300 250 200 150	800	
$V_{CE(sat)}$	Collector-Emitter Saturation Voltage	I_C = -500mA, I_B = -5.0mA I_C = -1.0A, I_B = -10mA I_C = -2.0A, I_B = -50mA		-0.25 -0.45 -0.75	mV
$V_{BE(sat)}$	Base-Emitter Saturation Voltage	I_C = -1.0A, I_B = -10mA		-1.0	V
$V_{BE(on)}$	Base-Emitter On Voltage	I_C = -1.0A, V_{CE} = -2.0V		-1.0	V
Small Signal Characteristics					
f_T	Transition Frequency	I_C = -50mA, V_{CE} = -5.0V, f = 50MHz	100		MHz

* Pulse Test: Pulse Width ≤300μs, Duty Cycle ≤ 2.0%

Thermal Characteristics

Symbol	Parameter	Max.	Units
P_D	Total Device Dissipation	2	W
$R_{\theta JA}$	Thermal Resistance, Junction to Ambient	62.5	°C/W

 Rev. B, November 2002

QSD2030F

FAIRCHILD SEMICONDUCTOR®

January 2008

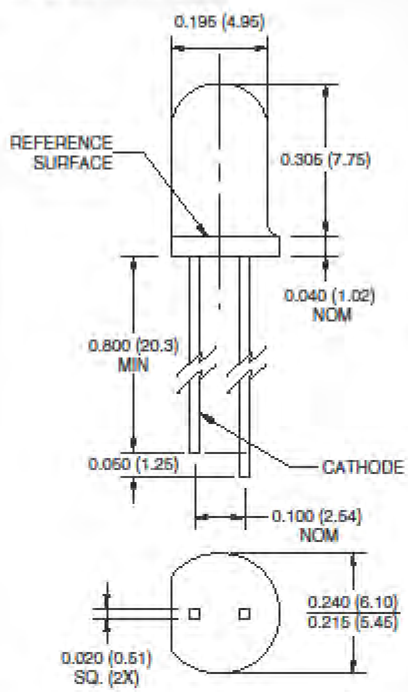
QSD2030F — Plastic Silicon Photodiode

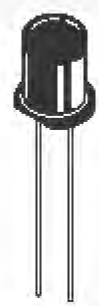
QSD2030F
Plastic Silicon Photodiode

Features

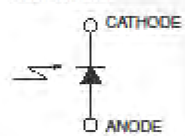
- PIN photodiode
- Package type: T-1 3/4 (5mm lens diameter)
- Wide reception angle, 40°
- Daylight filter
- Package material and color: black epoxy
- High sensitivity
- Peak sensitivity λ = 880nm
- Radiant sensitive area: 1mm x 1mm

Package Dimensions





Schematic



Notes:
1. Dimensions for all drawings are in inches (mm).
2. Tolerance of ±0.010 (0.25) on all non-nominal dimensions unless otherwise specified

QED223

FAIRCHILD SEMICONDUCTOR®

August 2008

QED221, QED222, QED223
Plastic Infrared Light Emitting Diode

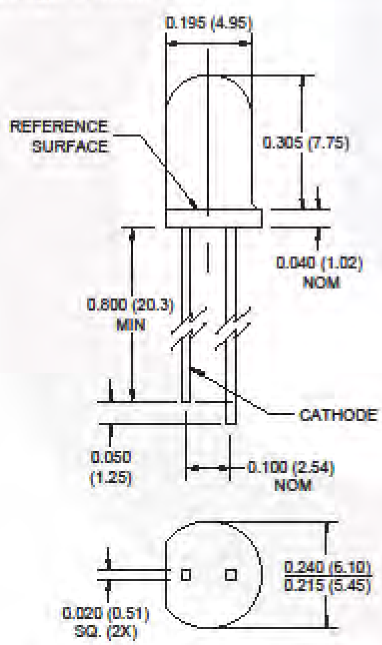
Features

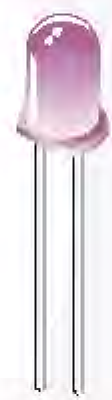
- λ = 880nm
- Chip material = AlGaAs
- Package type: T-1 3/4 (5mm lens diameter)
- Matched photosensor: QSD122/QSD123/QSD124
- Medium wide emission angle, 40°
- High output power
- Package material and color: clear, purple tinted, plastic

Description

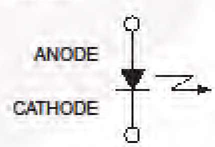
The QED221, QED222 and QED223 are 880nm AlGaAs LEDs encapsulated in a clear purple tinted, plastic T-1 3/4 package.

Package Dimensions





Schematic



Notes:

1. Dimensions of all drawings are in inches (mm).
2. Tolerance is ±0.010 (0.25) on all non-nominal dimensions unless otherwise specified.