# Property Protecting Turret

*Andrey Ivannikov*
*Eric Poon*
*Ilya Brutman*

**Mechatronics Project**

# Table of Contents

## The Drive, Our Purpose

The three of us work at a classified facility day to day. As such, this location is extremely strict on security. There are numerous sophisticated and expensive security systems in place, sometimes only to protect a single item. What makes these security systems expensive and different from any house alarm is the sensor sensitivity, sensor count, and installation costs. The same is also true of security systems in museums, etc.

In an attempt to streamline security, we went back to the drawing board to come up with a new scheme. Instead of surrounding the threat with the bait and security, what if we put security between the threat and bait? Instead of securing an area with hidden guards to protect a VIP, why not surround the VIP with bodyguards? This way, we can have a handful of bodyguards performing all the functions of the hidden guards, and do it better.

Using this model of security also addresses another downfall in traditional alarm systems: response time. The most prized alarm systems pride themselves on their response time, but they almost always alert the authorities and await response. Systems that have active defense are prohibitively expensive to install and implement.

Our system achieves all this in a single cost-effective unit. Taking inspiration from a security robot in the game Portal, we have implemented a turret which is able to passively detect motion, actively locate the originator, and preemptively defend anything in its coverage area.

Effectively something that is able to replace guardsmen, our product is also able to perform a wide range of functions where one would normally require non-discriminatory security personnel. Two such applications that come to mind are crowd control, and securing a demilitarized zone. This report will comprehensively and extensively

cover our prototype and what makes this possible, and how we've achieved it.

## Our Implementation

### *Design*

Our defense robot is essentially an intelligent turret. Prototyped from plywood and a plastic toy gun that we had on hand, this robot consists of 1 display, 1 pushbutton switch, 1 encoder/pushbutton combination switch, 1 key switch, 1 reed switch, 1 potentiometer, 1 relay, 1 set of Darlington Pair Transistors, 3 DC motors, 1 continuous servo, 2 non-continuous servos, 1 ultrasonic distance sensor, and 1 passive IR sensor. All this is connected with the proper electronics, powered by a 9.6V NiCd battery pack, and intelligently controlled by an Arduino Clone microcontroller.

The mechanical range of motion on our robot is about 185˚ panning and about 270˚ tilt. This motion is limited by bump stops to prevent damage to the servos used to move the turret around. In application, our turret has an operational panning range of 160˚, and a tilt range of 50˚. The base is low, sturdy, and wide to prevent toppling of the turret when in motion.

Our robot was built with safety and integrity in mind, both from the point of view of the user and the robot itself. With the exception of the emergency cut-off switch, a single servo, and the motion/distance sensors, everything is mounted on the moving platform. This allows for compact packaging and a more robust operation since it provides for minimum moving parts apparent to the microcontroller. In the event of accidental dislodging, the connectors for all parts will disconnect easily to prevent permanent or extensive damage. This also facilitates ease of maintenance. All breadboard connections go to a breakout board with connectors for everything. The 8 wires on the ribbon cable consist of (in order): regulated power, unregulated power, servo control, siren (not implemented, for future use), ground, motion sensor, ground, distance sensor. There are two grounds to isolate the motion sensor and the distance sensor from all the noisy signals. Unregulated power comes from the battery up top, goes to the key switch in the base, gets

regulated, and sent back to the breadboard. This allows the turret to be disconnected and reconnected to and from the base in a matter of 10 seconds for easy transport and setup.

The user interface consists of a 16x2 LCD display, a piezoelectric speaker, an encoder/button combination confirmation/selection knob, and a cancel/exit button. The LCD display is a white-on-black backlit unit that takes basic serial input. This allows for operation in the dark. The encoder/button combination switch is basically 3 switches. This switch has 5 legs. 2 of them constitute the basic legs of a pushbutton switch. The other 3 legs are Gnd, Sw1, Sw2. As the encoder is turned clockwise, (Sw1,Sw2) will read (0,0), (0,1), (1,1), (1,0), in order, repeating. When turned counter-clockwise, the order of (Sw1,Sw2) will reverse accordingly. Thus, monitoring the change in the readings of Sw1 and Sw2 will allow the microcontroller to determine if the user is rotating the knob left or right. All switches are wired to a 1kΩ pull-up resistor to have a steady reading at all times.

Actuation of the actual turret is realized by 2 servos: 1 continuous and 1 standard non-continuous. The panning servo is non-continuous, rigidly mounted by the servo-horn on the base attachment piece. The body of the servo is floated on a snug-fitting slot in the rotating turret base. The weight of the moving turret is supported by a “lazy-susan” ball-bearing. The tilt servo is of the continuous variety, but geared down to be able to move the relatively heavy gun. Because of the gearing, the internal potentiometer was replaced by an external one to report back to the microcontroller how far up/down it is tilting. This also allows for instantaneous feedback since we have direct access to the potentiometer. This tilt servo is a provision for improvement. As the turret stands now, it serves as a static tilt locator. With the addition of software, this can be an additional added feature. The servos are controlled via PWM output from the microcontroller.

The gun being used is a toy foam dart gun. The magazine is a rotating barrel of 20 foam darts, which is turned and then inserted into accelerators to quickly launch the dart forward. The actual motions of the gun is achieved via 3 DC motors. The trigger of the gun has been modified by removing the tensioner springs and adding barrel rotate

bumpers, then fitted with a DC Lego motor. As this motor pulls and pushes the trigger back and forth, the barrel is rotated and a foam dart is pushed into the two accelerator motors. When continuously fired, the entire barrel of 20 foam darts empties in about 6 seconds. The acceleration motors are switched by the microcontroller through Darlington Pair transistors, and the Lego motor is switched by the microcontroller through a relay. A reed switch for actuation feedback has also been fitted to the trigger to limit the amount of shots fired at each target.

Detection of motion is achieved via a passive IR sensor. This sensor has a parabolic dome which refracts light with an 180˚x180˚ hemisphere into the IR detector. When the amplitude of light changes, motion is detected. This then activates the target-locating ultrasonic sensor.

The ultrasonic sensor plays 2 parts in the operation of the turret. This analog sensor has a 20˚x20˚ cone, and can accurately detect the distance of objects up to 20' away, and only takes 50ms to report a reading. This sensor sits atop a non-continuous servo which pans 180˚ to detect any difference in the surrounding environment when motion is detected.

The microcontroller that we went with is called an Arduino Clone, which makes use of the Atmel ATmega328. This features USB connectivity, PWM hardware, onboard A-to-D, and C++ programmability. This microcontroller boasts 16MHz operation, internal pull-up resistors, built-in current limiters, 32kB of Flash memory, 2kB of SRAM, a 1kB EEPROM, and 20 I/O pins with built-in protection resistors: 14 digital, 6 analog. In case of faulty operation, the fitted key switch on the stationary base will remove power from the entire robot.

### *Operation*

Upon initial power-up, the robot enters standby mode, and the user is allowed to select the mode of operation of the robot. The possible modes of operation are selected with the knob, and the possible modes are Arm Turret, Set Scan Range, Set Number of Shots, Set Cool Down Period, and Test Fire. To Set Scan Range, the user simply rotates the turret to the left limit with the knob, and presses on the combination

knob to confirm, and repeat for the right limit. To Set Number of Shots, the user is prompted for how many shots to fire at a target. This is set using the knob, and a press on the knob will confirm it. Finally, to set the cool down period, the user will select the cool down interval where the gun will wait for the target to fall to the floor and stop moving before reacquiring another target. Test Fire mode will demonstrate the turret's ability to rapidly shoot the set number of shots. Arm Turret simply puts the turret in attack mode.

In attack mode, the distance sensor will first scan and map the room. After the initial scan, the robot will lay silent until motion is detected via the passive IR sensor. Once motion is detected, the distance sensor will activate and quickly scan the room to locate the changes. Once the changes are found past a certain threshold, the turret will take aim and fire. As a provisions for future improvement, code can be added to adjust the tilt of the gun to compensate for gravity and drag on the dart, relative to the distance of the object. The mechanics for this are already implemented.

## The Algorithm

The attack program has 2 basic subsections: Map Room, and Find Target.

### *Map Room*

The empty room is mapped over the course of 2 passes. The first pass is identical to the second pass. The maps from each pass are averaged together to generate the final map of the room. The servo rotates in steps to map out the room. For each step the ultrasonic distance sensor takes 4 distance measurements. Taking just one distance measurement proved too unreliable as the reported distance varies from measurement to measurement. Sound absorbent objects such as curtains and couches produce more variance then non-absorbent objects like walls. To filter this noise the average of the 4 distance measurements is taken. The amount the 4 distance vary, or in other words how "fuzzy" the measured object is, is also recorded as part of the room map. If the 4 measurements vary by too much then they are scraped and another 4 measurements are attempted.

### *Find Target*

To find the target a map of the room is made and compared to the empty room map. The two maps should be identical to each other except for one place where the distances are closer to the turret, a bulge in the map sort to say. That bulge is the target. The new map is analyzed on the fly as it is created. The newly acquired distance in the new map is immediately compared to the original distance in the empty room map. If the new distance is closer to the turret then the original distance then the target is found – no further scanning is needed. There is a threshold, of course, that the new distance needs to be lower by before it is considered to be different then the original map. That threshold depends on how fuzzy this location was in the original map. The fuzzier the object, the closer the new measurement needs to be in order to be identified as a target and not just natural variance in the distance measurement.

## The Product

*Parts Listing, Cost Analysis, and Rationale*

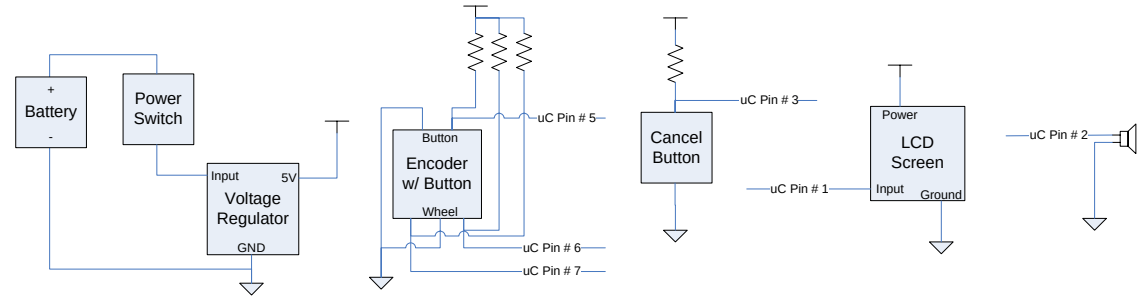
Here is the parts listing for our prototype. The final cost came out to $203.50. The most expensive items in this build are the servos, microcontroller, LCD Display, and the distance sensor. When purchasing in bulk of 1000 units, the cost will be much lower. Given that production units will use something like a BBgun and will require metal tooling for the chassis, we do not expect costs to exceed an estimated $50. After taking marketing overhead, packaging, warranty and support costs, and profit into consideration, these units can easily be sold for under $200. This is much more economic than any comparable security system out there, and does so at a fraction of the cost.

| Part | Cost | Qty | Subtotal | Description | Reason for use |
|---|---|---|---|---|---|
| Fairchild TIP 102 | $0.70 | 2 | $1.40 | Darlington Transistor | High Current output, low input current |
| Atmel ATmega328 | $25 | 1 | $25.00 | Microcontroller w/ board | Internal resistors, A2D, USB, C++ |
| Sparkfun LCD-00813 | $27 | 1 | $27.00 | LCD Screen | Easy to use, easy to read |
| Hanse Electronics SE- | $10 | 1 | $10.00 | Motion Sensor | Easy to use, just an open collector pin |

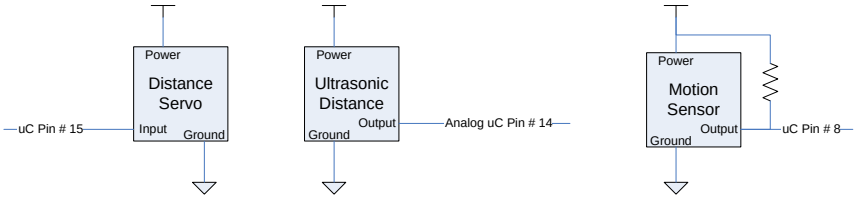
| | | | | | |
|---|---|---|---|---|---|
| 10 | | | | | |
| Panasonic EVQ-WTEF2515B | $1 | 1 | $1.00 | Rotary Encoder | Improves user interface |
| Maxbotix LV-EZ4 | $28 | 1 | $28.00 | Ulstrasonic Distance Sensor | Narrow beam, high accuracy |
| Hobbywing 5V/6V 3A UBEC | $10 | 1 | $10.00 | Voltage Regulator | Ease of use, high power, small package |
| Standard Servos | $13 | 3 | $39.00 | Servo | Built-in electronic feedback |
| Lego Motor | $18 | 1 | $18.00 | Motor | Internally geared |
| 9.6V NiCad Battery Pack | $10 | 1 | $10.00 | Battery pack | Small size, high instant power |
| Lazy Susan | $10 | 1 | $10.00 | Swivel bearing | Large size, ease of mounting |
| Radioshack 271-1721 | $3 | 1 | $3.00 | 10k Potentionmeter | Long shaft for easy mounting |
| Radioshack 273-074 | $3.50 | 1 | $3.50 | Piezo Speaker | User Interface |
| Plywood | $15.60 | 1 | $15.60 | Robot Chassis/Base | Easy Prototyping |
| Miscellaneous Buttons | $2.00 | 1 | $2.00 | Buttons | User Interface |
| Total | | | $203.50 | | |

# Our Resources – Technical Drawings, Diagrams, and Source Code

User Interface

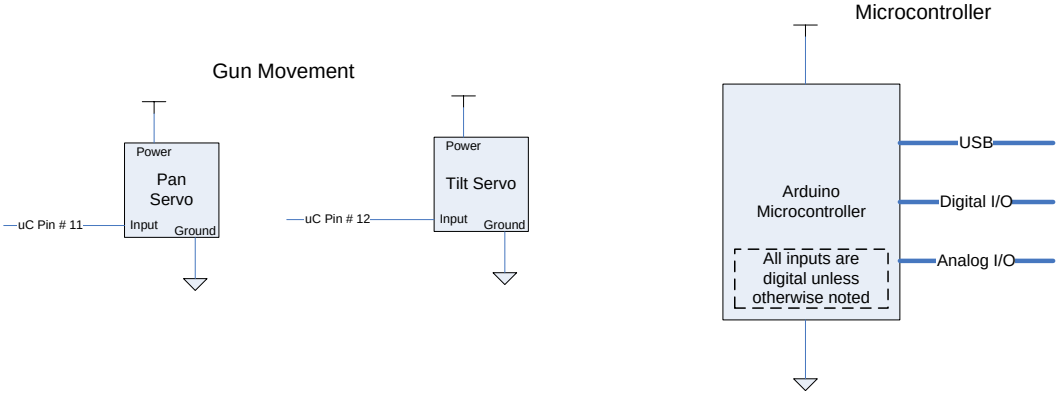



Scanning

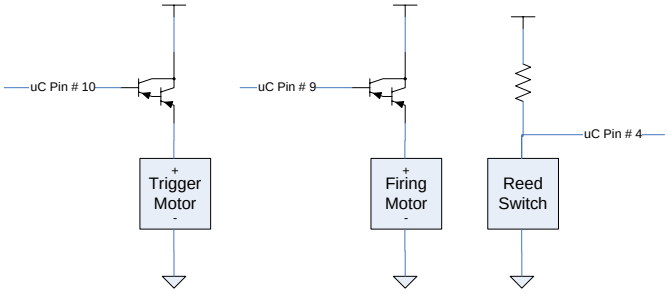


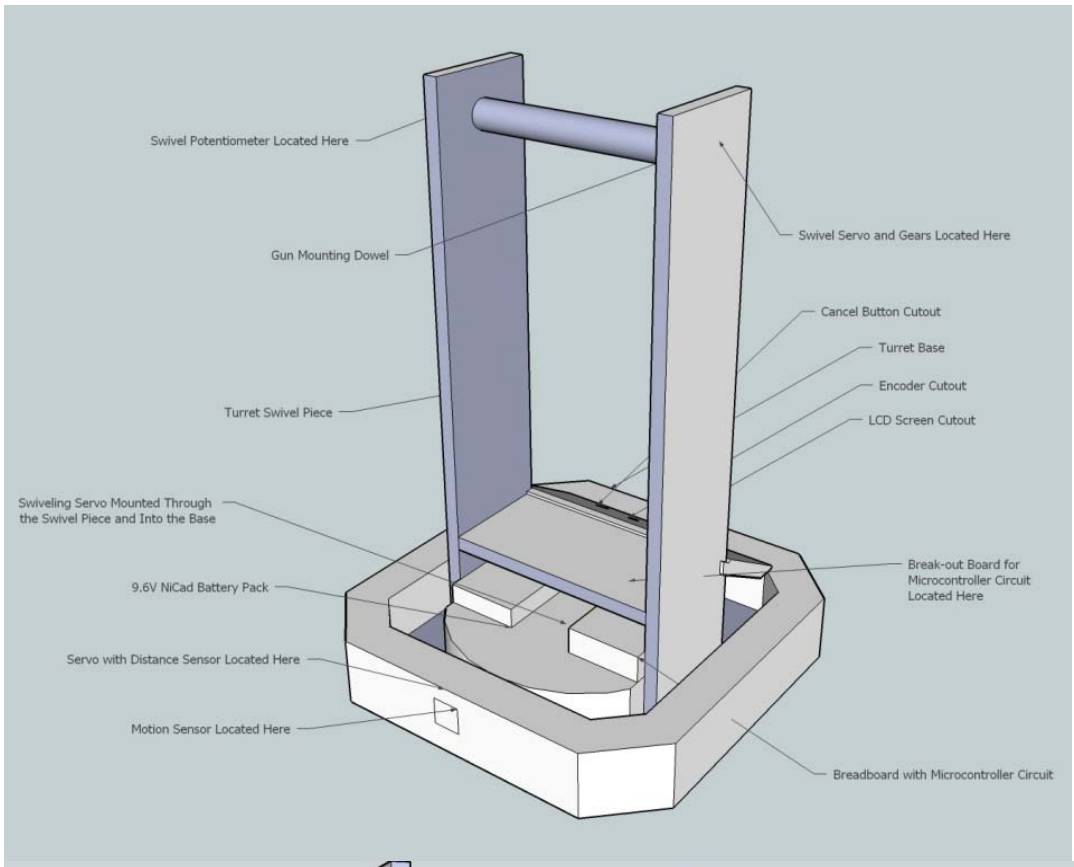

Microcontroller

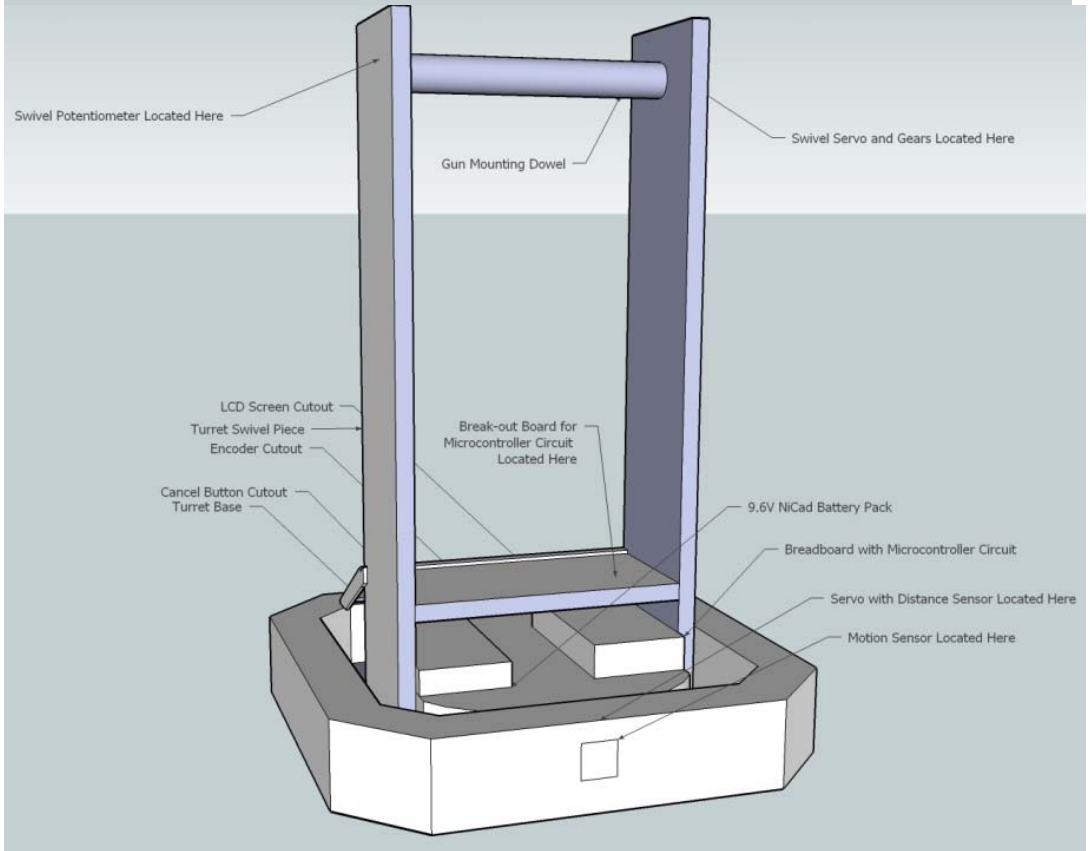
Gun Movement

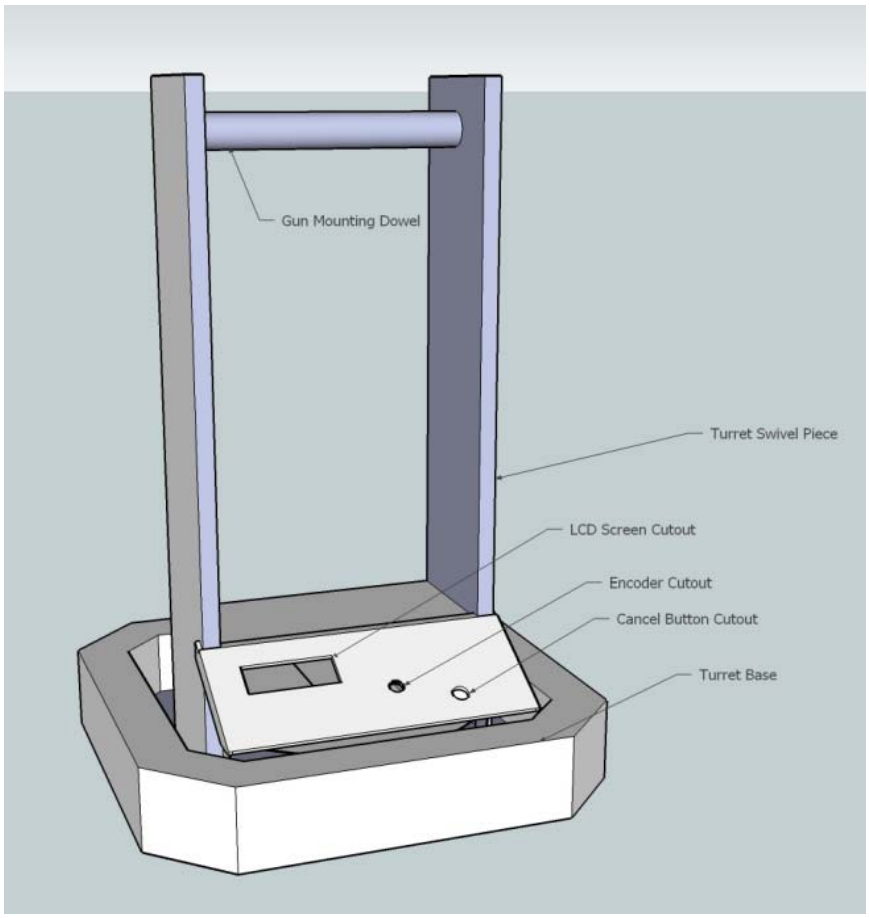


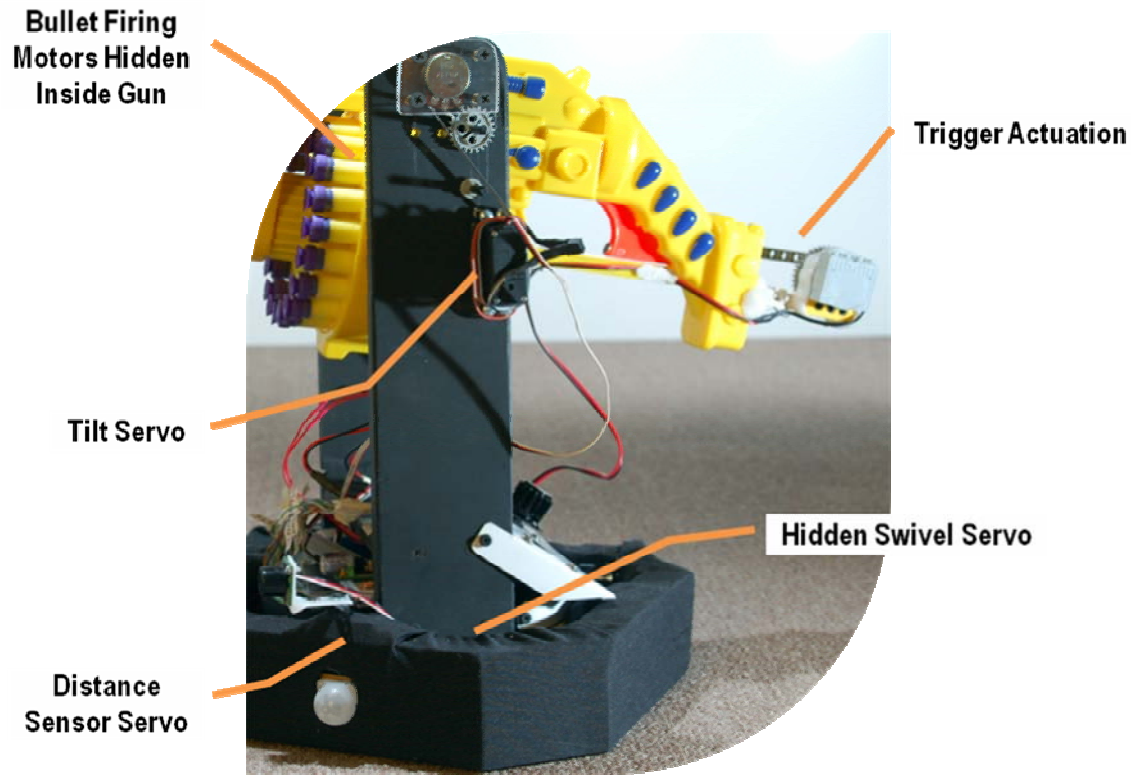

Gun Actuation

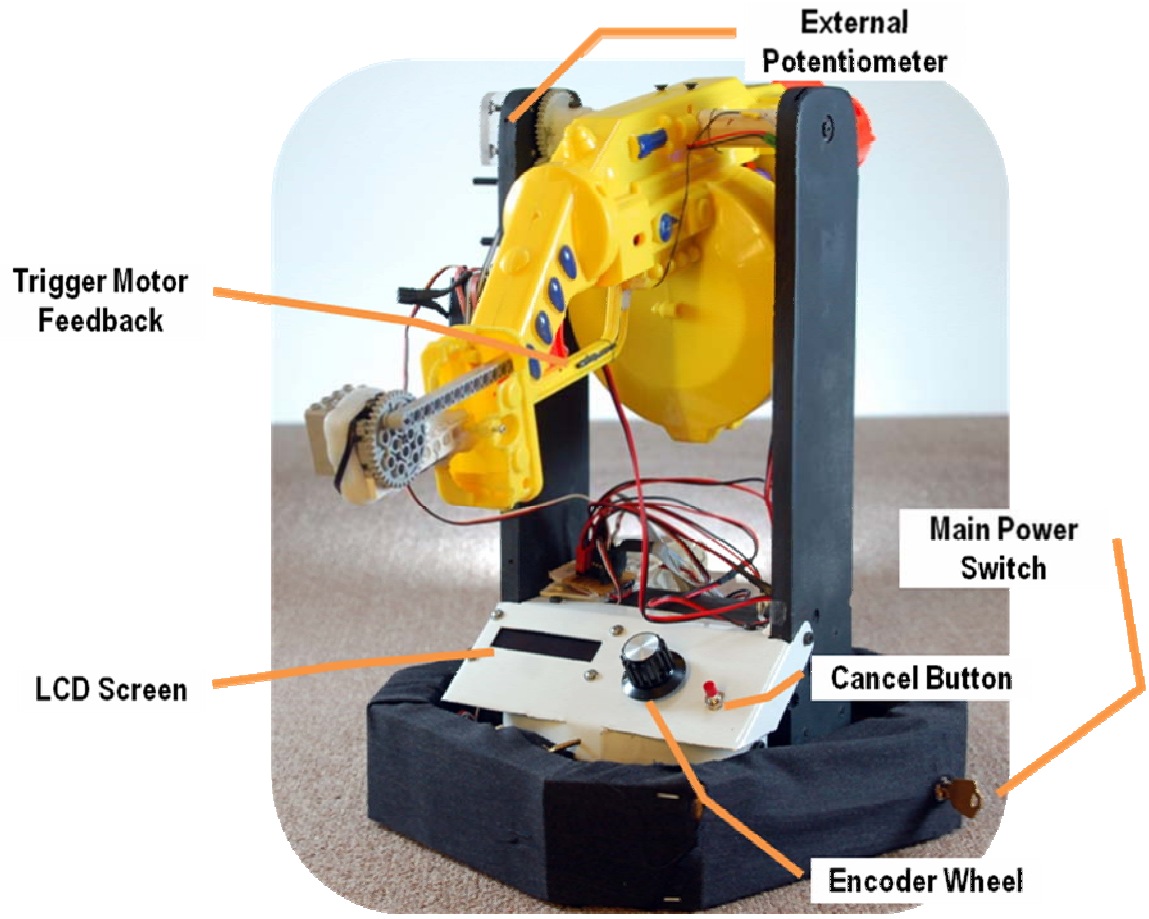


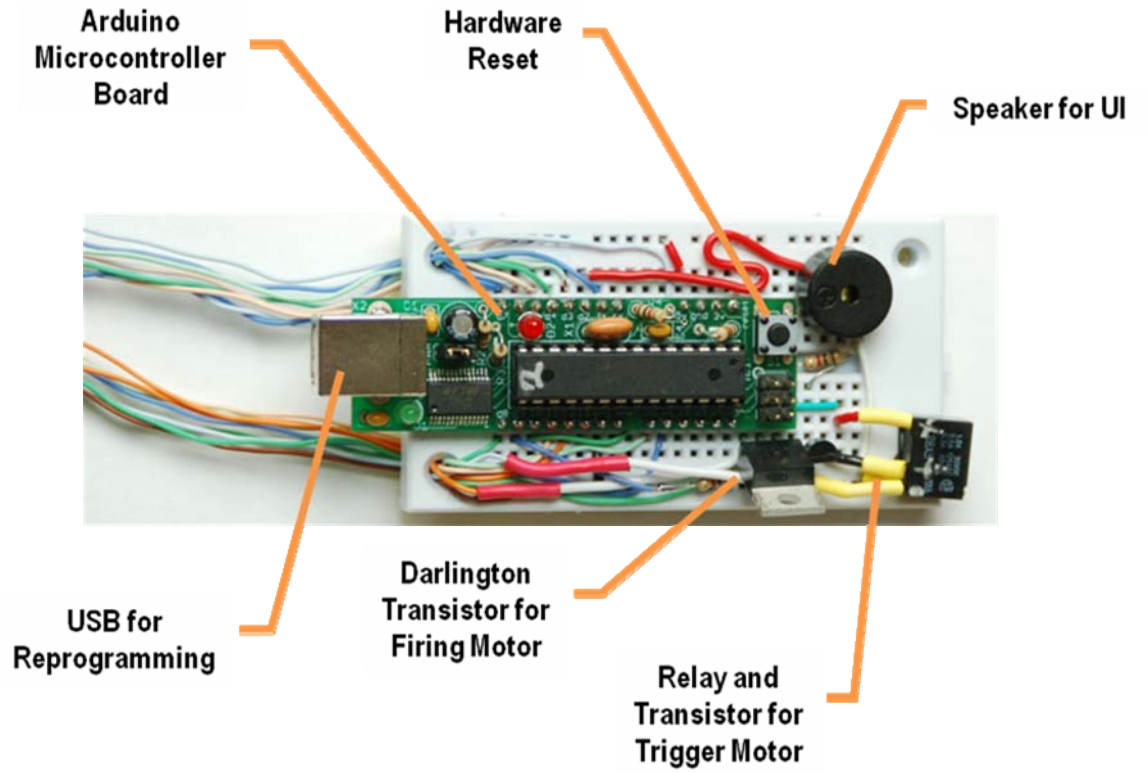

Swivel Potentiometer Located Here
Gun Mounting Dowel
Swivel Servo and Gears Located Here
Cancel Button Cutout
Turret Base
Encoder Cutout
LCD Screen Cutout
Turret Swivel Piece
Swiveling Servo Mounted Through the Swivel Piece and Into the Base
Break-out Board for Microcontroller Circuit Located Here
9.6V NiCad Battery Pack
Servo with Distance Sensor Located Here
Motion Sensor Located Here
Breadboard with Microcontroller Circuit

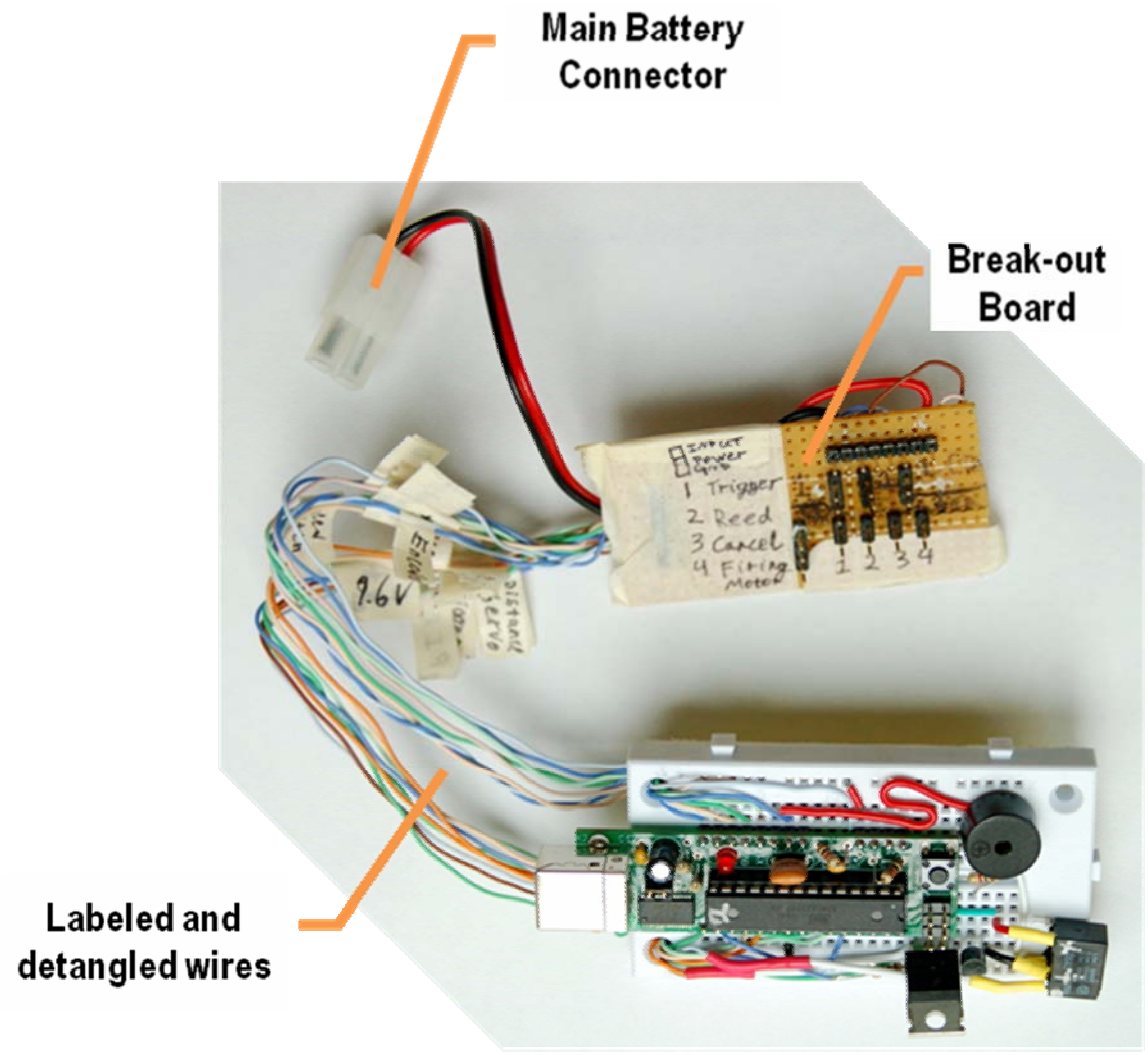

Swivel Potentiometer Located Here
Gun Mounting Dowel
Swivel Servo and Gears Located Here
LCD Screen Cutout
Turret Swivel Piece
Encoder Cutout
Cancel Button Cutout
Turret Base
Break-out Board for Microcontroller Circuit Located Here
9.6V NiCad Battery Pack
Breadboard with Microcontroller Circuit
Servo with Distance Sensor Located Here
Motion Sensor Located Here


Gun Mounting Dowel
Turret Swivel Piece
LCD Screen Cutout
Encoder Cutout
Cancel Button Cutout
Turret Base


Bullet Firing Motors Hidden Inside Gun
Trigger Actuation
Tilt Servo
Hidden Swivel Servo
Distance Sensor Servo


External Potentiometer
Trigger Motor Feedback
Main Power Switch
LCD Screen
Cancel Button
Encoder Wheel


Arduino Microcontroller Board
Hardware Reset
Speaker for UI
USB for Reprogramming
Darlington Transistor for Firing Motor
Relay and Transistor for Trigger Motor


Main Battery Connector
Break-out Board
Labeled and detangled wires
1 Trigger
2 Reed
3 Cancel
4 Firing Motor
1 2 3 4
9.6V