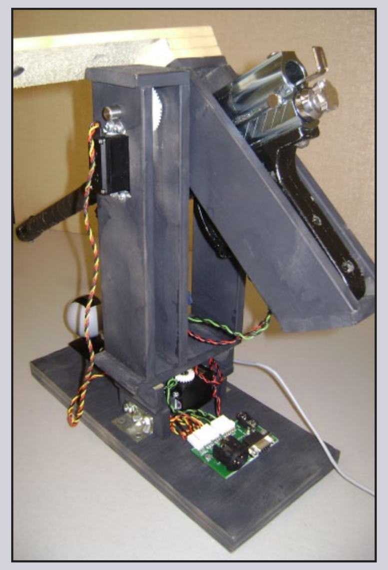
Team IC-U

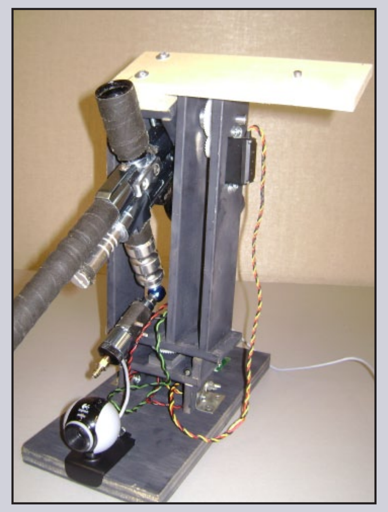
Intellegent Detection and Deterrence

N. Bosler, T. Florencki, O. Omusi, M. Wohlwend

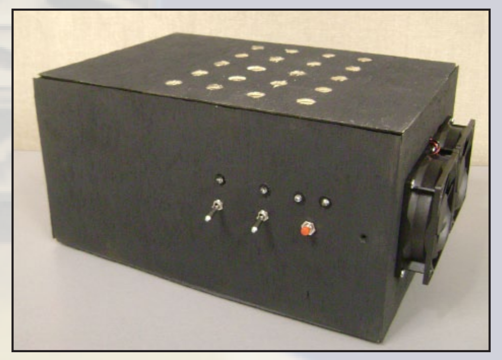
Mount Design



This mount provides a solid platform for the gun, high-res camera, and servo motors. It also places the rotating points near the radial center of gravity. The seeker cam is mounted to the front to provide a consistent point for calibration. The mount also includes the auxiliary board to route the signals from the power cable and data cable.



Case Design



The system case contains the computer board, power supply, hard drive, and microcontroller board. It is designed to contain all these components in a compact frame for ease of use. Two fans provide the necessary cooling to protect system components. The case is to be placed inside a building with the necessary cables routed outside to the mount. The system-on button, system-enable switch and gun-enable switch make up the simple user control located on the front of the case.

System Communication

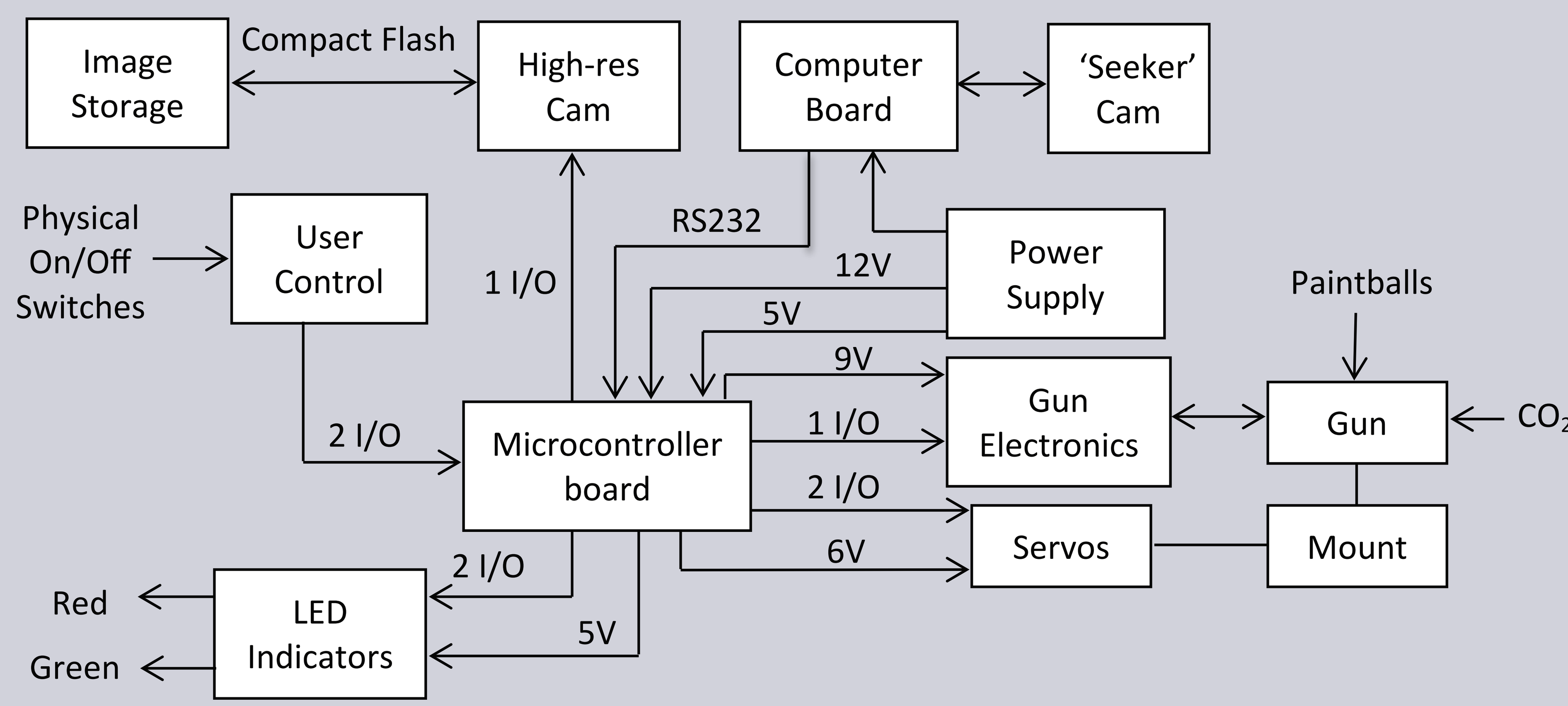
Pulse-width modulation is used to control the position of both servos. The duty cycle of the signal (T=20 ms) determines the angular position. Microcontroller code based on interrupt service routines is used to implement this.

RS232 communication between the computer and microcontroller is used to relay the position of the intruder (relative to the 'seeker' camera frame) and also to send commands to fire the high resolution camera and the gun. The microcontroller takes this information and sends the appropriate signals to the auxiliary board.

Future Enhancements

Numerous future enhancements can be made to this system.
- More processing power which will increase system responsiveness
- Wireless/Ethernet communication to provide a more integrated and a robust software user interface
- Mounting system not made of wood that will be strong enough to mount the camera and increase precision of servos
- Infrared/night vision web cameras to make the system relevent in an unlit nighttime situation

Block Diagram



System Requirements

-Accurate motion sensing algorithm in a somewhat constant background. This needs to output a definite 'target' to the overall system processor.

-VGA video output of 'seeker' camera to monitor motion sensing algorithm in real time. The VGA output will allow for an external monitor to be connected to the system so that the user may see the input from the 'seeker' camera in addition to 'target selection' from the motion following algorithm.

-Motion of gun/hi-res camera tied to location of detected motion from incoming video feed. The system processor takes the 'target' as defined by the motion sensing algorithm and aligns the gun to that 'target'.

-Trigger control of the gun though the overall system processor when motion is detected and 'target' position is correct.

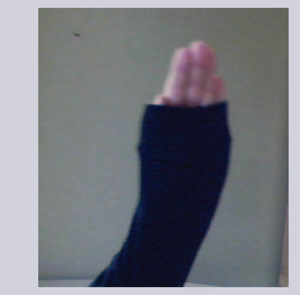
-Control of hi-res camera image capture. The images being output from the hi-res camera need to be captured and saved using either the overall system memory or its own dedicated memory.

-Removable Digital storage of hi-res camera images. Images saved by the camera system need to be permanently stored on removable storage to be easily accessible to the user.

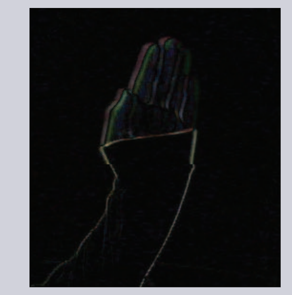
-Ability to toggle the use of high resolution image capture and paintball gun fire. This functionality allows the user to set their desired security parameters and decide which of the two features to enable.

Motion Algorithm

The motion algorithm compares the current frame to the previous frame and determines the changes. The orignal method compared the current frame to the running average of previous frames, but this was changed in favor of the faster frame-per-frame comparison. The code is further explained below using examples.



Initally, a frame is captured to determine frame size and to set previous image for comparison.



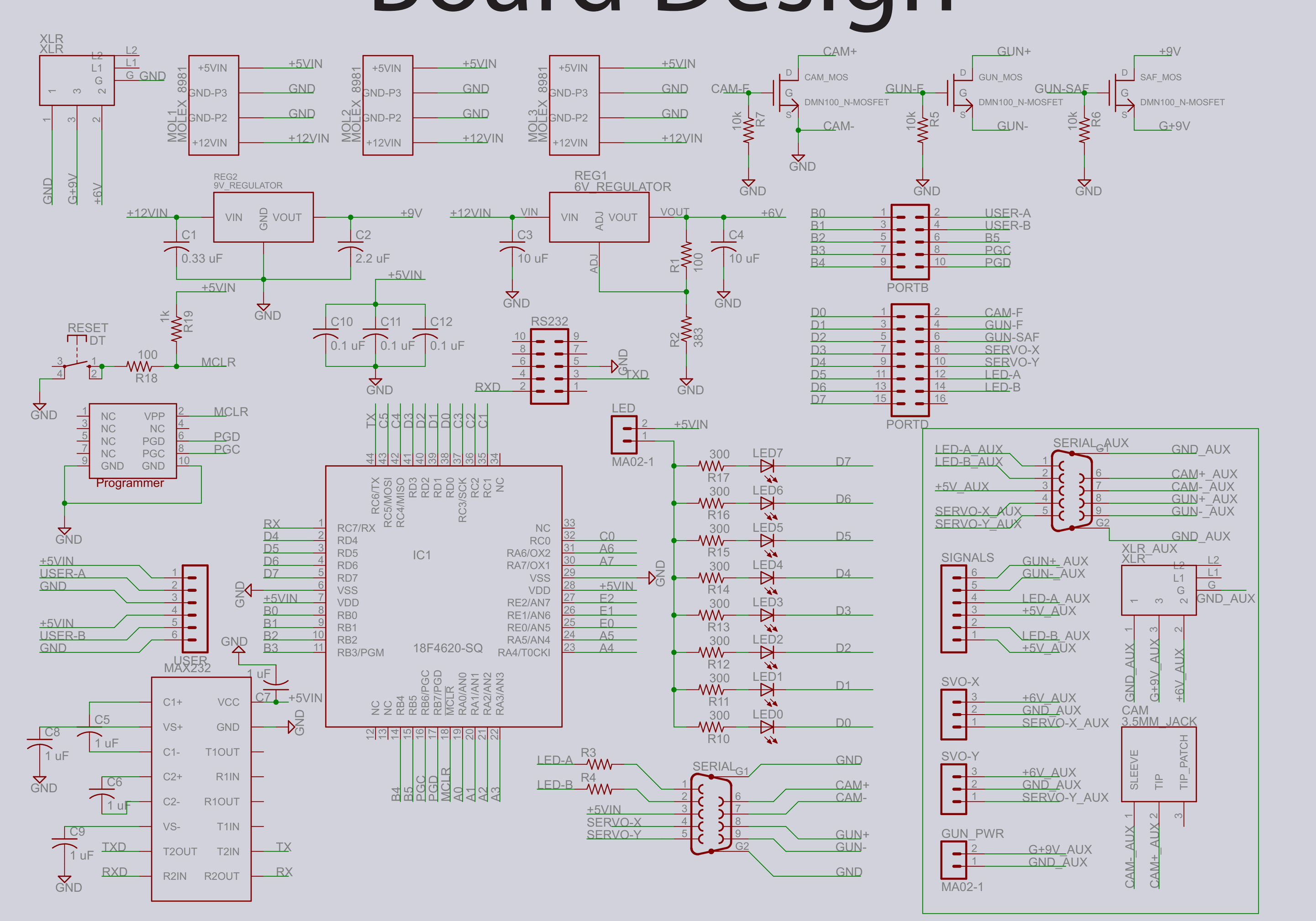
The current image is then subtracted from the previous image to determine the difference between frames. This difference represents the motion detected.

The difference image is converted to greyscale and then to black and white in order for the contour fuctions to work. White pixels in this image represent the areas of motion.

This black and white image is then used to draw contours around the white pixles of movement. Rectangles are drawn around connected contours. These rectangles give local center points of movement. The area of these rectangles is compared to a threshold to remove noise.

The final step is to average all the centers into a single point that is then sent to the microcontroller as the servo positions. Before the values are sent to the microcontroller, the center is converted from the (X,Y) plane to the corresponding (Φ,θ) angles.

Board Design



The schematic shows the connections for both the main board and auxiliary board (outlined in green). An XLR and D-sub 9 cables connected the main and auxiliary board.

EE Senior Design 2011

University of Notre Dame