Security Center

ECE 511 Project Report

Ibraheem Darab  
David Pachowicz

Jeremy Barthélemy

Cam Duong

Security Center Final Report

December 2012

Table of Contents

[Abstract 3](#_Toc343461764)

[Motivation 4](#_Toc343461765)

[Solution 4](#_Toc343461766)

[Interface 6](#_Toc343461767)

[Hardware 6](#_Toc343461768)

[MSP430 LaunchPad 6](#_Toc343461769)

[XBEE OEM RF Wireless Module 6](#_Toc343461770)

[Servo 6](#_Toc343461771)

[Motion Sensors 6](#_Toc343461772)

[Sound Sensor 6](#_Toc343461773)

[Push Buttons 6](#_Toc343461774)

[Software 6](#_Toc343461775)

[UART 6](#_Toc343461776)

[Timer 7](#_Toc343461777)

[GPIO 7](#_Toc343461778)

[Results and Conclusions 8](#_Toc343461779)

[Appendix 9](#_Toc343461780)

[Parts List 9](#_Toc343461782)

[Schematic 11](#_Toc343461783)

# Abstract

Existing commercial security cameras are limited, with no automated tracking. The goal was to construct a security center capable of providing visual monitoring of an area of interest. The security center incorporates multiple sensors and a command system to provide direction to the camera control system, thus making it a smart system sensitive to both motion and sound while also providing the operator with wireless management, monitoring, and controls.

# Motivation

With the limited offerings of today’s surveillance systems, many users are looking for a system that provides automated capabilities, remote access, control, and reporting. Home users would feel more secure knowing their home is being monitored, by receiving real time alerts. Commercial users can benefit from a system that can save them security personnel labor costs, networking bandwidth, and storage of high definition video.

# Solution

The Security Center system brings to the market an adaptable surveillance package that will work with any wireless camera currently on the market. The package includes a rotating mount for the camera with tracking sensors, and a wireless remote. The package is split into two modules: the System Command and the System Controller.

The System Command is held by the user monitoring the target area. It sends commands and receives status from the System Controller via long range ZigBee wireless communication. Users have the option of manually controlling the camera or enabling the System Command to make decisions for auto-panning.

The System Controller mounts and pans the user installed camera. Motion and sound sensors mounted within the enclosure provide telemetry within the room. The System Controller reports this data back to the System Command. It pans the camera via a servo motor depending on commands received by the System Command.



1. Figure 1 – Block Diagram
2. Figure 2 – Communication Diagram

# Interface

## Hardware

### MSP430 LaunchPad

Each System Module used one LaunchPad. Both used the MSP430G2553 chip. The System Command used one button on the LaunchPad for auto/manual mode and used one LED for the alarm.

### XBEE OEM RF Wireless Module

Each System Module used one XBEE module, and communicated to the XBEE module using dedicated hardware UART tx and rx pins. The XBEE modules used 2.4GHz Zigbee 802.11.15 protocol. The module runs off of +3.3V.

### Servo

The servo was connected to the System Controller LaunchPad via a GPIO. A PWM signal was used to control the signal. The servo runs off of +6V.

### Motion Sensors

Each of the three motion sensors were connected to the System Controller LaunchPad via GPIO with pull down resistors. The single pin output goes high when motion is detected, and returns low when no motion is detected.

### Sound Sensor

The sound sensor module is connected to the System Controller LaunchPad via GPIO. It has a potentiometer to modify the sensitivity. The single pin output goes high when sound is detected, and returns low when not sound is detected.

### Push Buttons

The push buttons were used on the System Command to send commands to the System Controller. They were interfaced to the LaunchPad via GPIO and pull up resistors.

## Software

### UART

The UART registers were used to communicate between the MSP and the XBEE module. It was configured to 9600 baud, 8 data bits, 1 stop bit, no parity. The receive buffer was interrupt based, while the transmit buffer was filled as needed.

### Timer

Timer A was used to generate the PWM signal for the servo. The CCR0 register set the period to 20ms, and the CCR1 register set the pulse width was set between 1.0 and 2.0 ms. The timer was turned on only for the duration of the servo motion to conserve power consumed by the servo.

### GPIO

General Purpose IOs were used for handling the pushbutton switches, motion sensors, and sound sensor. The switches were interrupt based, and the motion and sound sensors were polled.

# Results and Conclusions

The implementation was a success. The system was able to detect the presence of motion and sound in the target area. It was able to successfully track a person walk around in a 12x12 room. Button presses successfully paned the camera from a remote location. Setting the sound sensor threshold very high successfully removed unnecessary alarms due to background noise. Narrowing the field of view of the motion sensors successfully made them more directional to assist in proper automatic panning of the camera.

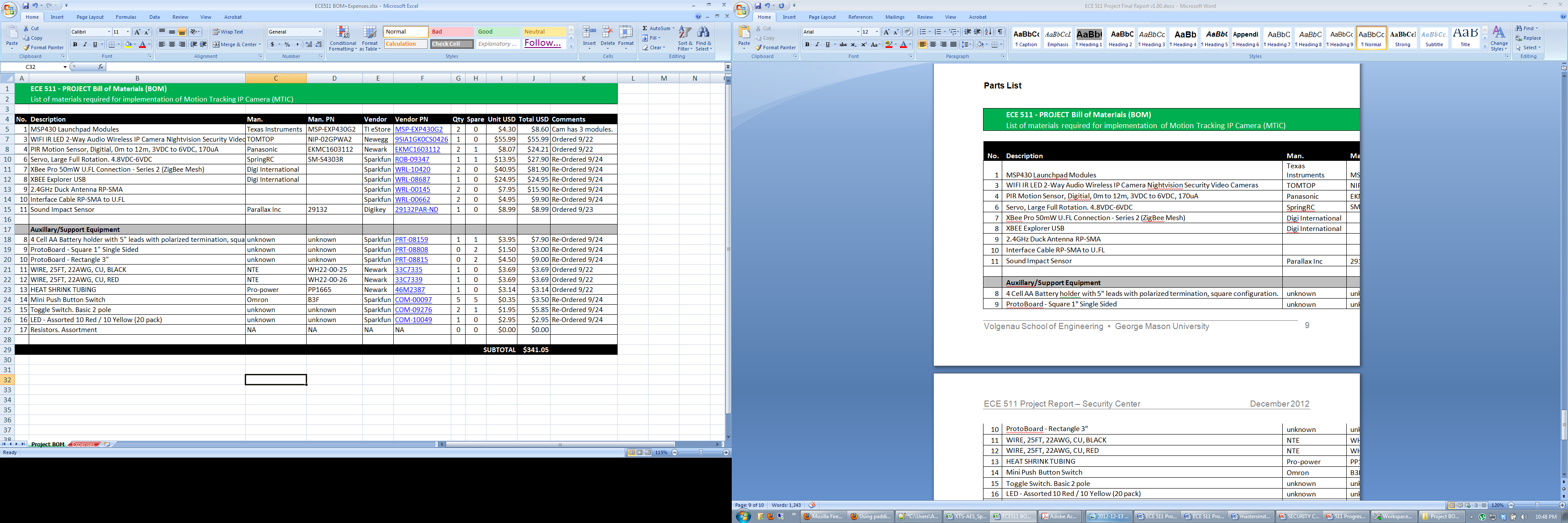
One lesson learned was to back up the work on a regular basis. Even the slightest changes can leave you lost the next day.

Another lesson learned was to comment the code. This helped with debugging, understanding registers, and understanding functions.

# Appendix

## Parts List

See below



## Schematic

