SCADA Home Automation

By
Jonathan Beason, Ben Curths, Simone Gbouomou
Ben McAnulty, Chad Bryan, Ben Calvert

CPE 495 Computer Engineering Design I
CPE 488 Cybersecurity Engineering Capstone I
Electrical and Computer Engineering
The University of Alabama in Huntsville

Table of Contents

Meet The Team

Cybersecurity Engineering

Ben Calvert

Simone Gbouomou

Jon Beason

Ben McAnulty

Computer Engineering

Ben Curths

Chad Bryan

Project Summary

The SCADA Home Automation System is a physical and interactive education model designed to emulate today's smart home systems. The purpose of the project is to introduce and generate interest in cybersecurity among students and young professionals through the use of an interactive model. By creating an interactive model, the students and young professionals will be able to see the effects of cyber-attacks manifest in a physical manner. To accomplish the project objectives, a solenoid, stepper motor, and sensors will be integrated using Arduino microcontrollers running OpenPLC. In addition, a Human Machine Interface (HMI) will be used to control and monitor the SCADA Home Automation System. Students and young professionals will have the opportunity to interact with the system through physical interactions with the model and by performing cyber-attacks on the system.

Reason For Project

- Create an interactive model that can be used for educational purposes.
- Generate interest in Cybersecurity/Computer Engineering amongst students and young professionals.
- Reduce the shortage of Cybersecurity/Computer Engineering professionals.
- Demonstrate why Cybersecurity/Computer Engineering is important.
- Promote UAH by highlighting skills/knowledge gained through attending UAH’s College of Engineering.

Marketing Requirements 1

Flashy:

Multiple changing lights, makes sounds including alarms or buzzer, and looks visually interesting.

Interactive:

Moveable components with reactive feedback.

Portable:

Lightweight, Ergonomic, Easy to set up, and Simple to use

Marketing Requirements 2

Technical:

SCADA system built using OpenPLC, Modbus and Ladder Logic. Monitor the system using an Human Machine Interface (HMI).

Hacks:

Hollywood style movie hacks, Physically/Visually changes the state of the product. Non-technical style hacks that are easy to understand/implement.

Patchable:

Easy to patch or mitigate the vulnerability of the SCADA system.

SCADA Home Automation Model

SCADA Home Automation Model(Smart Home control panel)

Components

Left Panel

- Model Home
- Smart Lock Mechanism
- Working garage door

Right Panel

- Smart Lock Control
- Garage Door Control
- Window Control
- Infrared Sensor
- Motion Sensor

System Design Description

- The deliverable will be a portable case that contains a model home with multiple physical security controls commonly found in modern households.
- These security controls will send their current states to a HMI displayed in the case.
- During a demonstration of our device, cyber attacks will be conducted which allow unauthorized access to the model home.
- These attacks will be conducted from an external computer connected through a network hub attached to the case.

Project Update

Completed Tasks

Project requirement & specification defined.

Selected components & determined layout.

OpenPlc install on Raspberry Pi 4.

Preliminary scaled model constructed.

Phase II Goals

Acquire materials & components
Construct case & model
Integrate components
Program system using Ladder Logic

Perform general testing
Cybersecurity analysis
Develop hacks
Acceptance tests

Testing Plan

Unit Testing: the functionality of the individual system components will be tested during the development of the Ladder Logic.

Integration Testing: after the system components are integrated into the model the components will be tested to ensure they work.

Regression Testing: if the system is altered by modification or the integration on new components the system will be retested.

Acceptance Testing: upon completion of the project the system as a whole will be tested to ensure it functions per the stakeholders request.

Project Management Plan

Scrum-Based Agile Framework:

Modified Scrum-based plan consisting of 2-week-long sprints for maximum flexibility.

Regular Meetings:

Sprint kickoff and retrospective meetings for each sprint. Project sponsors/mentors are encouraged to attend to provide feedback, but this is not required. Additional mid-sprint meeting for communication and progress updates. Unscheduled irregular meetings will also occur as needed.

Peer Review and Accountability:

Anonymous peer review surveys collected after every two sprints. Contributions may be reviewed and project sponsor/mentor feedback may be collected.

Timeline (1)

ID	Activity	Description	Deliverable	Duration (hr)	People	Resources	Predecessors
1	**Project Initiation**						
1.1	Inventory	Check the Components on hand	Inventory List	1	Jon Beason		
1.2	Development Environment	Install OpenPLC on Rasp. Pi and Arduino		2	Jon Beason	Pi, Arduino OpenPLC	
2	**Case Construction**						
2.1	Prep Material	Measure / Cut material to specification		5	Ben Calvert, Ben Mcanulty	Saw , Tape Measure	
2.2	Sharp Edges	Check for sharp edges and file		2	Ben Calvert, Ben Mcanulty	Metal File	2.1
2.3	Assemble Case	Fastens case together		5	Ben Calvert, Ben Mcanulty	Hex keys	2.2
2.4	Structural Test	Test the durability of the case and adjust accordingly	Completed Case	3	Ben Calvert, Ben Mcanulty		2.3
3	**Building Model**						
3.1	Prep Material	Measure / Cut material to specification		10	Ben Calvert, Ben Mcanulty	Foam board, razor blade	
3.2	Assemble Model	Glue the foam board together using hot glue.		20	Ben Calvert, Ben Mcanulty	Dual temp. hot glue gun	3.1
3.3	Finish Model	Paint and add details to model	Completed Model	10	Ben Calvert, Ben Mcanulty	Paint	3.2
4	**Smart Lock System**						
4.1	Simulate Deadbolt	Create deadbolt using solenoid Program it with ladder logic	Deadbolt Prototype (DB prototype)	5	Chad Bryan, Ben Curths	Solenoid, Arduino, OpenPLC	
4.2	Combine smart lock and DB prototype	Integrate smart lock to work with the Deadbolt Prototype	Deadbolt Subsystem	15	Chad Bryan, Ben Curths	Smart lock	4.1

Timeline (2)

4.5	Integration Testing Deadbolt	Test the deadbolt system to ensure it still works correctly	Completed Deadbolt Subsystem	5	Chad Bryan, Ben Curths		4.4
5	**Garage Door System**						
5.1	Simulate Garage Door	Create garage door using DC motor and program it with ladder logic	Garage Door Prototype (GD prototype)	5	Chad Bryan, Ben Curths	DC Motor, Arduino, OpenPLC	
5.2	Garage remote and GD prototype	Integrate remote to work with the garage subsystem	Garage Door Subsystem	15	Chad Bryan, Ben Curths	Garage Remote	5.1
5.3	GD Unit Test	Test Garage Door system to ensure it works correctly		5	Chad Bryan, Ben Curths		5.2
5.4	Integrate Garage Door subsystem	Integrate GB subsystem into the building model		5	Chad Bryan, Ben Curths	Building model	3.2, 5.2
5.5	Integration Testing Garage Door	Test the Garage Door system to ensure it still works correctly	Completed Garage Door Subsystem	5	Chad Bryan, Ben Curths		5.4
6	**Window System**						
6.1	Window Prototype	Connect magnetic sensor to Arduino and program it with ladder logic	Window prototype	10	Jon Beason, Simone Gbouomou	Mag sensor, OpenPLC, Arduino	
6.2	Window Unit Test	Test the window subsystem to ensure it works correctly		1.5	Jon Beason, Simone Gbouomou		6.1
6.3	Window system Integration	Integrate window system into the model		1	Jon Beason, Simone Gbouomou	Building model	3.2, 6.2
6.4	Integration Testing Window system	Test the window system to ensure it still works correctly	Completed Window System	3	Jon Beason, Simone Gbouomou		6.3
7	**Motion System**						
7.1	Motion Prototype	Connect motion sensor to Arduino and program it with ladder logic	Motion prototype	10	Ben Calvert, Ben McAnulty	Motion sensor, OpenPLC, Arduino	
7.2	Motion Unit Test	Test the motion subsystem to ensure it works correctly		1.5	Ben Calvert, Ben McAnulty		7.1
7.3	Motion system Integration	Integrate motion system into the model		1	Ben Calvert, Ben McAnulty	Building model	3.2, 7.2

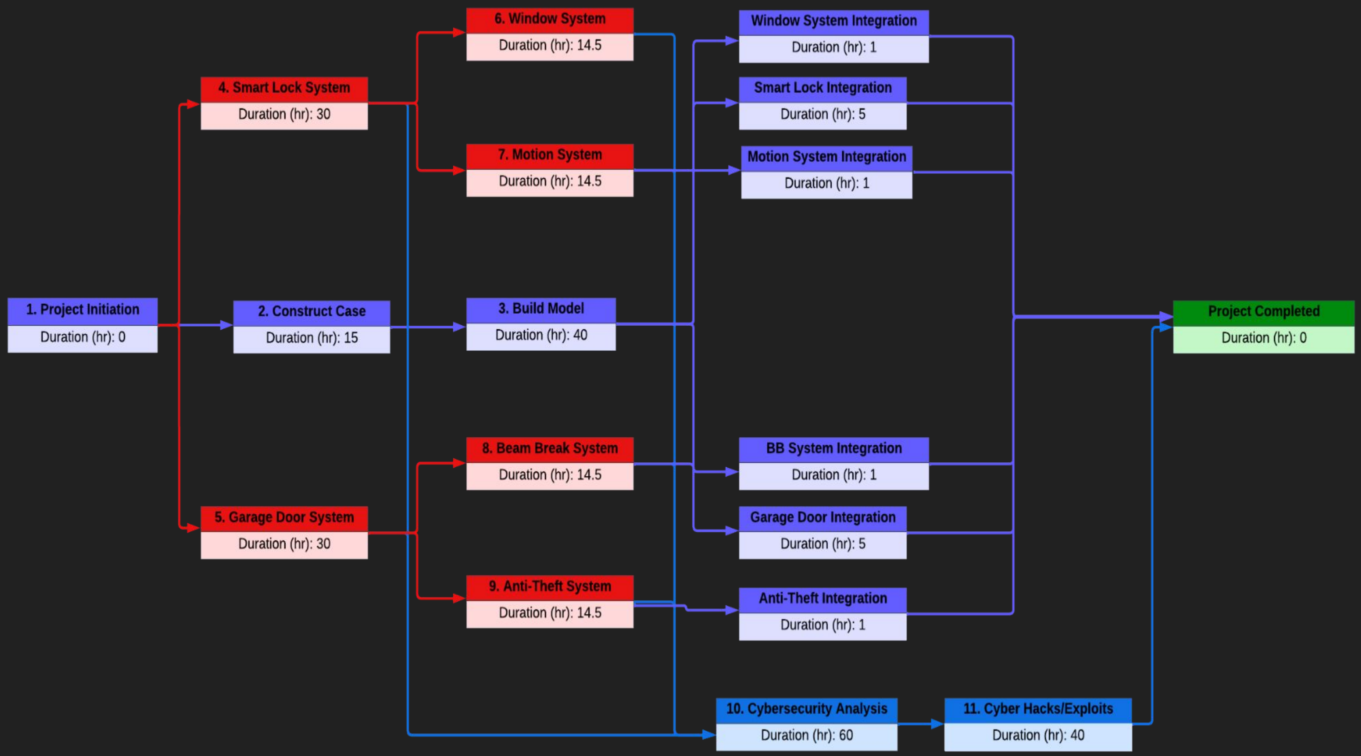
Timeline (3)

7.4	Integration Testing Motion system	Test the motion system to ensure it still works correctly	Completed Motion System	3	Ben Calvert, Ben McAnulty		7.3
8	**IR Sensor (Beam Break) System**						
8.1	Beam Break (BB) Prototype	Connect IR sensor to Arduino and program it with ladder logic	Beam Break prototype	10	Chad Bryan, Ben Curths	IR sensor, OpenPLC, Arduino	
8.2	BB Unit Test	Test the BB subsystem to ensure it works correctly		1.5	Chad Bryan, Ben Curths		8.1
8.3	BB system Integration	Integrate BB system into the model		1	Chad Bryan, Ben Curths	Building model	3.2, 8.2
8.4	Integration Testing Motion system	Test the BB system to ensure it still works correctly	Completed BB System	3	Chad Bryan, Ben Curths		8.3
9	**Anti-Theft System**						
9.1	Anti-Theft Prototype	Connect pressure sensor to Arduino and program it with ladder logic	Anti-Theft prototype	10	Chad Bryan, Ben Curths	Pressure sensor, OpenPLC, Arduino	
9.2	Anti-Theft Unit Test	Test the Anti-Theft subsystem to ensure it works correctly		1.5	Chad Bryan, Ben Curths		9.1
9.3	Anti-Theft system Integration	Integrate Anti-Theft system into the model		1	Chad Bryan, Ben Curths	Building model	3.2, 9.2
9.4	Integration Testing Anti-Theft system	Test the Anti-Theft system to ensure it still works correctly	Completed Anti-Theft System	3	Chad Bryan, Ben Curths		9.3
10	**Cybersecurity Analysis**						
10.1	Risk Assessment	Perform Risk Analysis on the system	Assessment Report	15	Jon Beason, Ben Calvert, Simone Gbouomou, Ben McAnulty		9.3
10.2	Threat identification	Research and Identify attack vectors	Threat Identification Report	15	Jon Beason, Ben Calvert, Simone Gbouomou, Ben McAnulty		10.1
10.3	Threat Mitigation	Create solutions to reduce/mitigate risk.	Mitigation Report	30	Jon Beason, Ben Calvert, Simone Gbouomou, Ben McAnulty		10.2

Timeline (4)

11	**Cyber Hacks/Exploits**						
11.1	Hack system	Develop attacks using the information from Threat Identification	2 System Exploits	35	Jon Beason, Ben Calvert, Simone Gbouomou, Ben McAnulty		10.1
11.2	Test Hacks	Test that hacks work correctly on the system	Completed System Hacks	5	Jon Beason, Ben Calvert, Simone Gbouomou, Ben McAnulty		11.1

PERT Chart



1. Project Initiation
Duration (hr): 0
4. Smart Lock System
Duration (hr): 30
2. Construct Case
Duration (hr): 15
5. Garage Door System
Duration (hr): 30
6. Window System
Duration (hr): 14.5
7. Motion System
Duration (hr): 14.5
3. Build Model
Duration (hr): 40
8. Beam Break System
Duration (hr): 14.5
9. Anti-Theft System
Duration (hr): 14.5
Window System Integration
Duration (hr): 1
Smart Lock Integration
Duration (hr): 5
Motion System Integration
Duration (hr): 1
BB System Integration
Duration (hr): 1
Garage Door Integration
Duration (hr): 5
Anti-Theft Integration
Duration (hr): 1
10. Cybersecurity Analysis
Duration (hr): 60
11. Cyber Hacks/Exploits
Duration (hr): 40
Project Completed
Duration (hr): 0

Division of Responsibility

Ben Calvert

Case/model Construction, Motion System, Cybersecurity Analysis, Cyber Hacks/Exploits

Ben Mcanulty

Model Construction, Motion System, Cybersecurity Analysis, Cyber Hacks/Exploits

Jonathan Beason

Project Initiation, Window System, Cybersecurity Analysis, Cyber Hacks/Exploits

Simone Gbouomou

Model construction, Motion system, cybersecurity analysis, cyber Hack/Exploits

Ben Curths

Smart Lock System, Garage Door System, IR Sensor System, Anti-Theft System

Chad Bryan

Smart lock system, garage door system, IR system sensor, Anti-theft system

Safety Analysis

Construction Phase

<u>Soldering Iron Burns</u>

Mitigation - designate a hot zone for soldering and label with warning signs.

<u>Toxic fumes from Solder Iron</u>

Mitigation - use lead-free soldering wire and solder in well ventilated areas.

<u>Electrical shock from prototyping</u>

Mitigation - disconnect power when modifying / connecting components.

<u>Sharp edges / tools during case construction</u>

Mitigation - wear PPE file sharp edges. Unplug tools when not in use.

Cost Analysis 1

Funding provided by the ECE Department:	$400
Estimated labor cost in hours:	287 hours
Itemized price and expenses:	$407
Fixed Cost:	None
Variable Cost:	Labor/Materials

A break down of all cost associated with the project can be seen on Cost Analysis 2

Cost Analysis 2

Components / Hardware

2	Raspberry Pi 4	$90
1	Arduino Mega	$50
1	Monitor	$70
2	3mm IR Sensors	$6
2	PIR Motion Sensor	$20
2	Magnetic Sensor	$8
1	Force-Sensitive Sensor	$6
1	Small Solenoid	$8
2	Micro Limit Switch 10pc	$7
1	DC Motor	$13
1	10pc 1000mm Alum. Extrus	$80
1	Smart Deadbolt	$40
1	5 Port Ethernet Hub	$10
	Subtotal	$407

Software / Services

OpenPLC Software	Free
scadaBr/ Home - Assistant	Free
Linux OS	Free

Labor Cost

Building Model	40 hours
Building Case	15 hours
Ladder Logic Programming	60 hours
Circuit Prototyping	20 hours
Integration	14 hours
Testing	38 hours
Cybersecurity Analysis	60 hours
Cybersecurity Hacks	40 hours
Total	287 hours

Project Q/A