

Ministry of Education and Science of the Russian Federation
Peter the Great St. Petersburg State Polytechnic University
Institute of Computer Sciences and Technologies
Graduate School of Cyber-Physical Systems and Control

Course Project

Cloud Service Controller for SCADA Systems

Discipline: Industrial Automation and Drive Systems

25 May 2017

Student Group: 13541/8

Christopher W. Blake

Hubert Truchan

Professor

Potekhin V. V.

St. Petersburg
2017

Contents

Summary	3
Earlier Work Review	3
Interface	3
Text Version.....	3
Image Version.....	4
Background Code	4
Current Issues.....	4
Desired Improvements.....	5
System Overview	5
Web Interface.....	6
Mobile Compatibility.....	6
Monitor Page.....	7
SCADA Settings Page	7
Hardware Configuration Page	8
Documentation Page.....	9
Accounts Page	10
SCADA Controller	11
Downloading	11
Installation.....	11
Interface	11
Hardware API	12
Real-time Hub	12
Identity Server.....	13
Future Considerations	13
Conclusion	13

Summary

An existing SCADA system is being controlled via a prototype Windows-Forms-based software which provides proof-of-concept testing. Building upon this earlier work, it is now desirable to extend control of this SCADA system and others to the web.

A cloud service has been created using MVC ASP.Net (C#) to provide a desktop- and mobile-friendly website with various supportive services. Using this website, a user is able to be authenticated, create a SCADA control system, add hardware, and produce animated interfaces. Additionally, the service is structured to allow for future upgradability and extensions.

Earlier Work Review

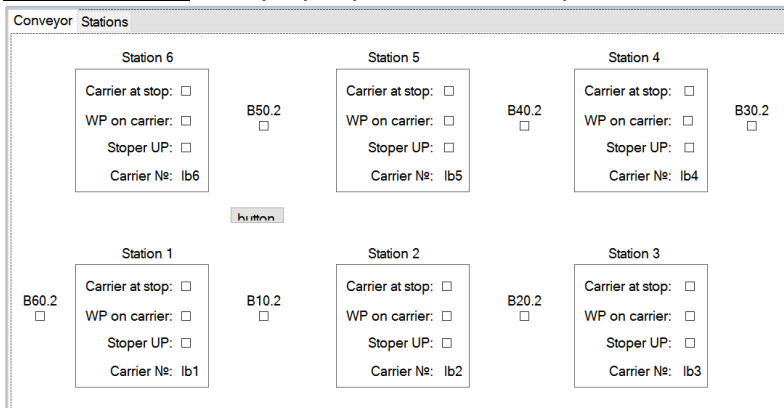
Interface

The earlier work consists of two different versions to produce graphical interfaces. A simple text-only version and an image supported version. There is however, no animation on either.

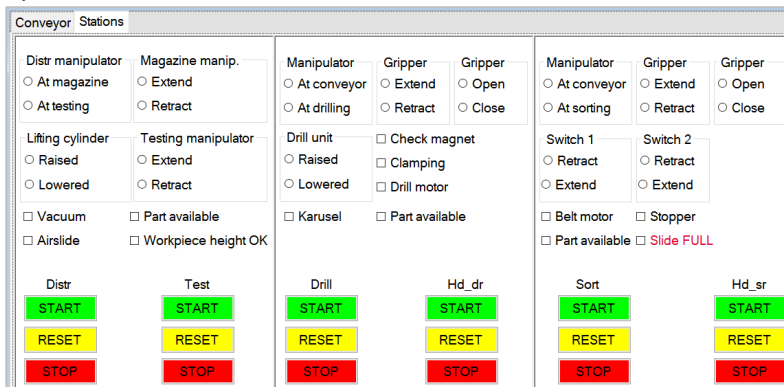
Text Version

The text version has two tabs and one command bar. It is essentially a series of organized labels with checkboxes or radio buttons, which show “checked” if the appropriate sensor is activated.

Tab: Conveyor – Displays updates as items pass around the conveyor belt.



Tab: Stations – Allows starting, resetting, and stopping of individual stations. It also shows updates from certain sensors.



Command Bar

The command bar only has a few operations, which combine commands from the different stations. For example, to turn everything on. However, these commands are not reliable.

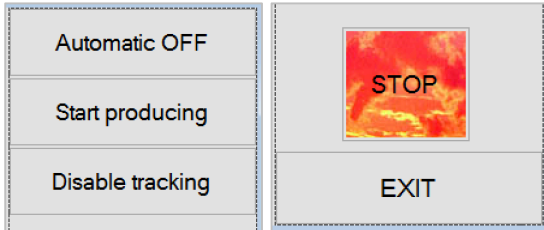
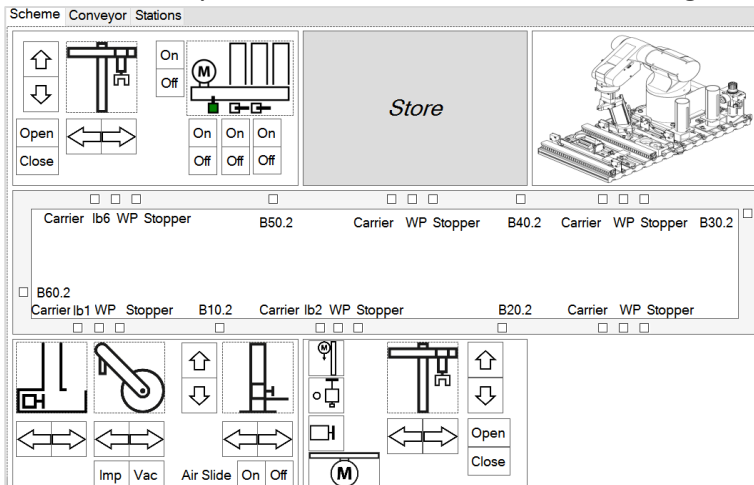


Image Version

The image version is an extension of the previous the text-based version. The extension simply adds a tab with images and check boxes. Some elements change color to represent status.

Tab: Scheme - A third tab was created in the graphical version. This tab attempts to implement both the conveyor and station data into one more logical view.



Background Code

The background code is fairly-well organized. However, it is not designed with future extensibility in mind. There is 1 overall master controller and 7 hardware classes to represent different hardware stations. There also appear to be some “testing” files which serve no purposes. Below is a list of the classes. Unfortunately, no documentation is available about which class relates to which hardware.

- | | |
|---------------|---------|
| 1. Controller | 5. HdDr |
| 2. Conveyor | 6. HdSr |
| 3. Dist | 7. Sort |
| 4. Dril | |

Current Issues

1. Documentation – There is no information available except for sparse code comments.
2. Laggy Interface – Many commands run in the interface thread; no async processes.
3. Commenting – There exists no or very few code comments.
4. Acronyms – Many non-intuitive acronyms are used for variables and object names.
5. Static Structure – All elements are statically defined inline.

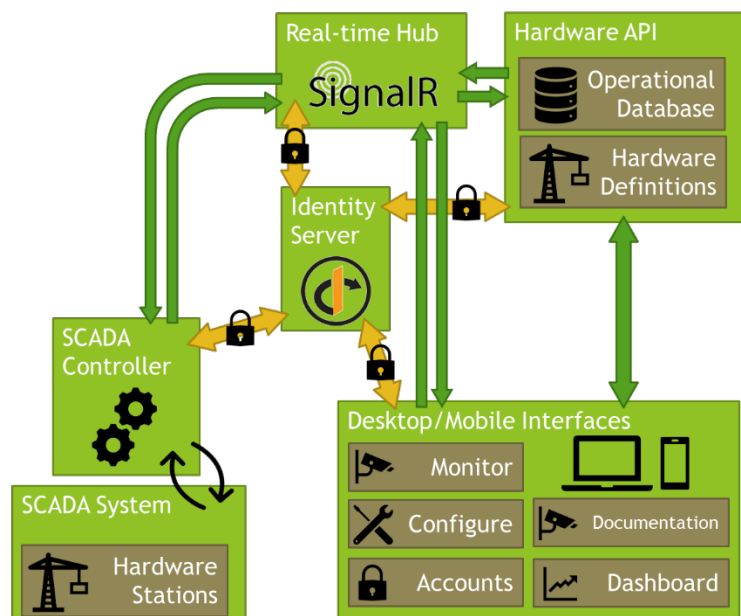
Desired Improvements

Having reviewed the existing design, the following improvements are desired or believed possible.

1. Code documentation
 - a. Source code should have regular commenting.
 - b. Final report should contain structures and processes.
2. Upgradeable design pattern
 - a. Web API service for each modular component. Examples:
 - i. Web Interface
 - ii. SCADA Hardware
 - iii. Real-time Communication
 - iv. Account Management
 - b. Dynamically add available elements during runtime.
 - i. Adding/removing interfaces
 - ii. Adding/removing hardware
3. Web-based graphical interface
 - a. Javascript, CSS, HTML5 (no special plugins such as flash, java, ...)
 - b. Modular controls to represent hardware.
 - c. Future upgradeable to add additional animations or display information.
4. Secure Online Portal
 - a. A secure and reliable system with proven history should be utilized.
5. Operation data buffered into a database.
 - a. Buffer size can be adjusted.
 - b. Supports future data mining projects.
6. Full documentation.
 - a. Program flow.
 - b. Acronyms should be clarified.
 - c. Software usage.

System Overview

The cloud service is separated into five components. Each of these works together to provide a seamless interface for controlling the SCADA system.

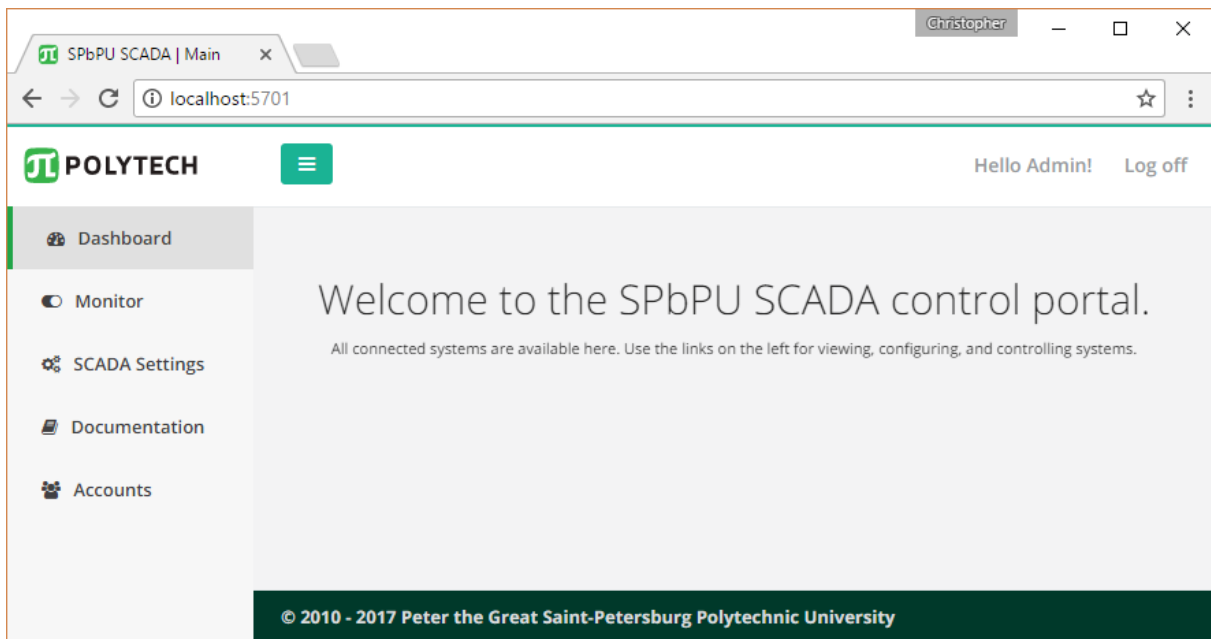


1. **Web Interface** – The main interactive interface for the user.
2. **SCADA Controller** – Windows service that controls and monitors the SCADA system.
3. **Hardware API** – The communication between clients and SCADA Controllers.
4. **Real-time Hub** – Enables push communications between all services.
5. **Identity Server** – Authentication of clients and issuing of access tokens.

Web Interface

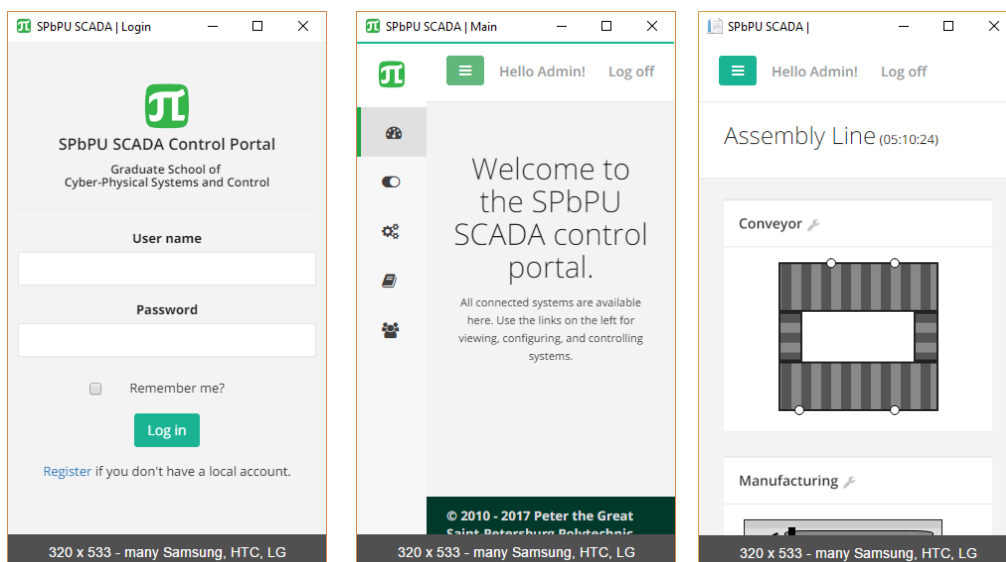
The web interface is the main method for user interaction. It provides interfaces for monitoring and controlling any number of SCADA-based systems. It additionally allows for the creation of and modification of existing systems in real-time. The interface is split into four sections. Available functionality is dependent on the user's permissions, described later.

- 1.) **Dashboard** – Serves as a homepage and future site for “overview” information about the systems.
- 2.) **Monitor** – Viewing and Controlling of an individual SCADA System.
- 3.) **SCADA Settings** – Adding and removing of systems, hardware, and data channels.
- 4.) **Accounts** – Assigning of roles to registered users.



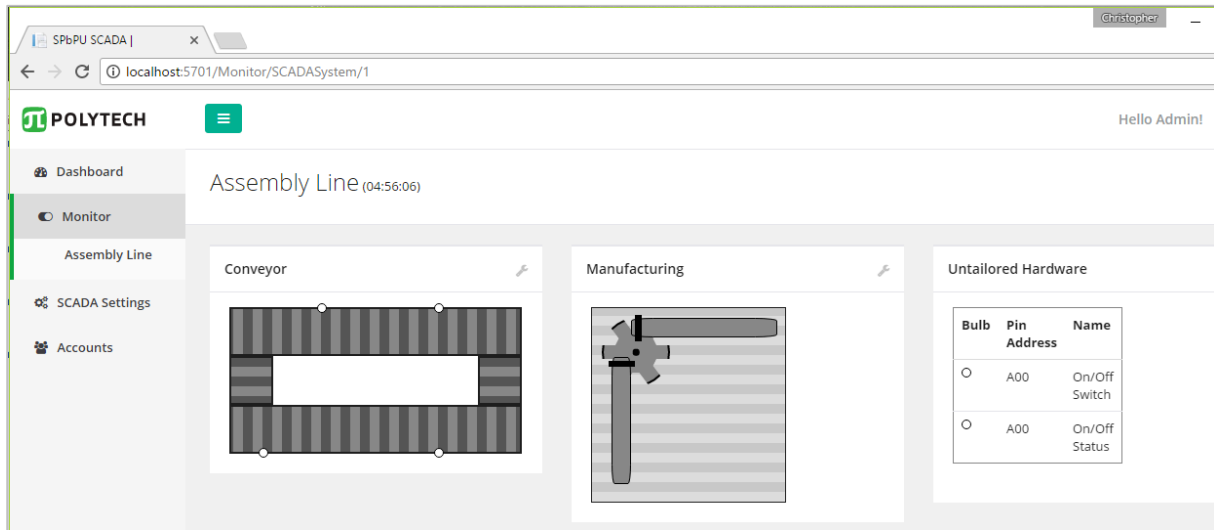
Mobile Compatibility

The entire site is built in a dynamic way. As such, all or most web pages should automatically format for mobile devices. Below are examples of the login, home, and monitor pages.



Monitor Page

The monitor page allows viewing of a single SCADA system. Each hardware component of the system is shown individually and realtime updates from the system are animated.



SCADA Settings Page

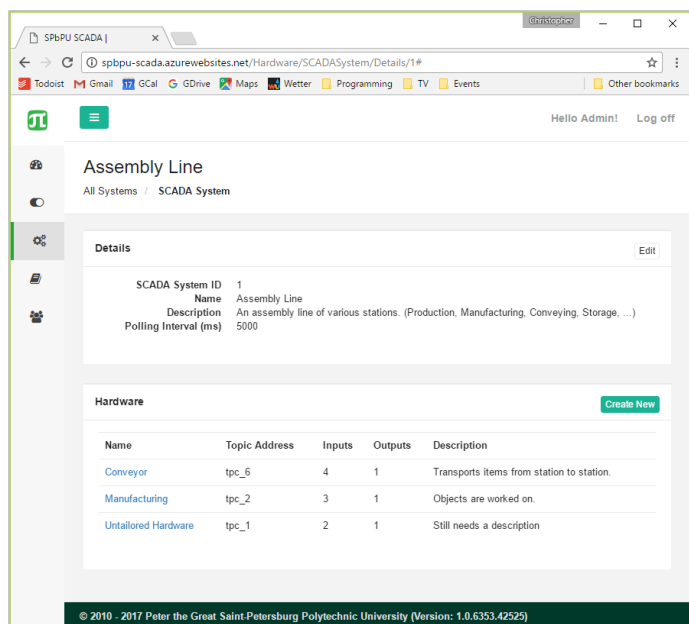
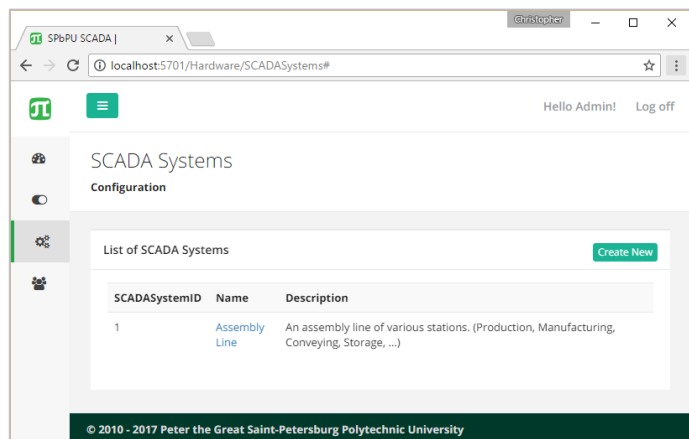
All current SCADA systems are shown on this page. Clicking the name of a system opens the details page for that system. A new system may be added by pressing the "Create New" button.

After selecting the SCADA system, the details of that system may be modified.

Available Fields:

- Name
- Description
- Hardware
- Polling Interval

Each hardware item for the SCADA system is listed as well. Clicking the name of a hardware item will show its configuration page. A new hardware item may be created using the "Create New" button.

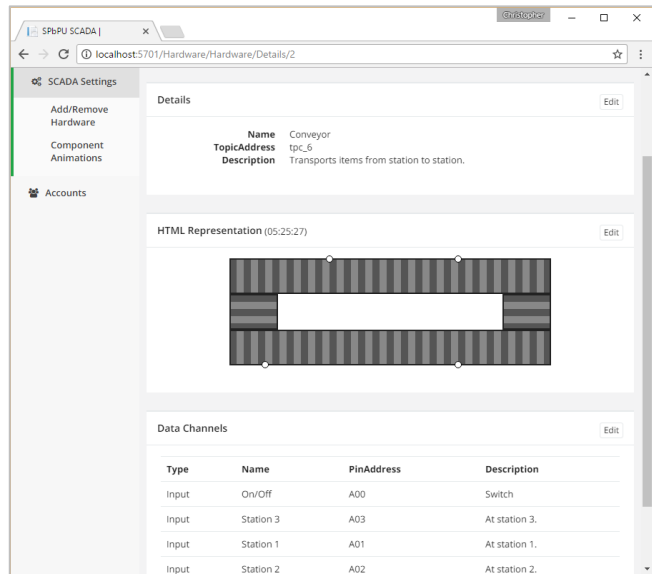


Hardware Configuration Page

After a piece of hardware is chosen, it can be configured. Each section may be modified by pressing the “Edit” button in the top right of the section.

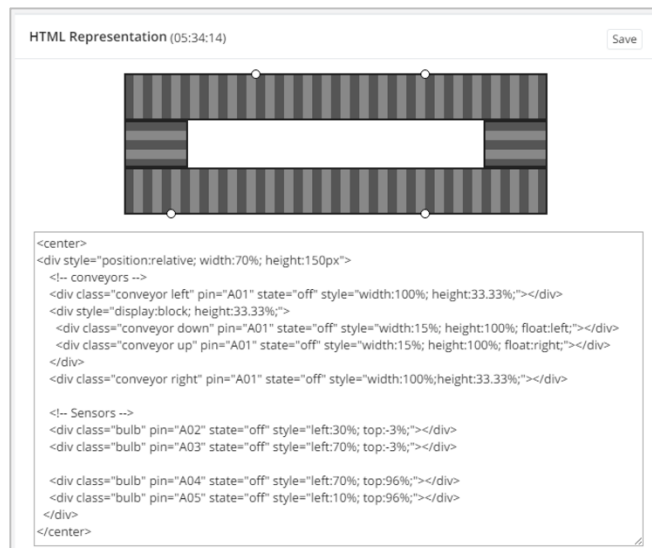
Available Fields:

- Name
- Topic Address
- Description
- HTML Representation
- Data Channels
 - Type
 - Name
 - Pin Address
 - Description



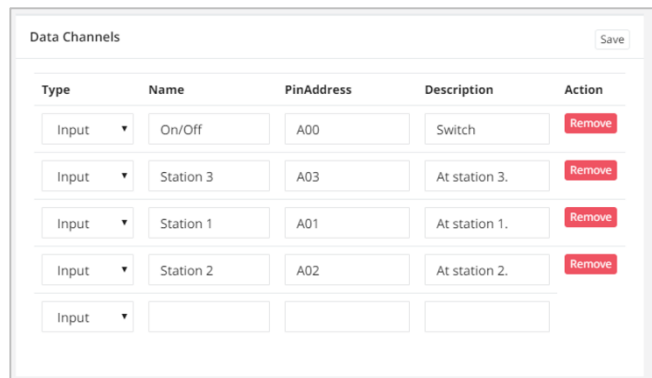
HTML Editor

The graphical representation of this piece of hardware can be designed using HTML and animated using CSS. As the user edits the html box, a live preview is shown above. If the system is active at the time, the affects of the sytem are also displayed.



Data Channels

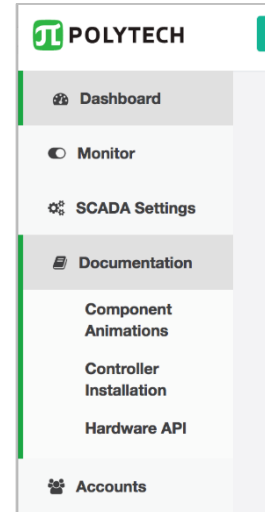
Channels and their associated pin address may be added, removed, or edited at any time. Upon saving, these changes are updated in the database and pushed to the local controller.



Documentation Page

Documentation for site usage is provided on the website. There are three sections, each of which describes a major topic. Below are screenshots for reference.

- **Component Animations** – Using pre-built animations to represent hardware.
- **Controller Installation** – Setting up a local computer to connect to the service.
- **Hardware API** – (admin only) Information about back-end API.



Conveyors

Conveyors can be used to show movement. There exist four options and they can be turned on/off;

- conveyor up
- conveyor right
- conveyor down
- conveyor left

Animation	Description	Code
	Conveyor Up	<code><div class="conveyor up" pin='A01' state='on'></div></code>
	Conveyor Right	<code><div class="conveyor right" pin='A01' state='on'></div></code>
	Conveyor Down	<code><div class="conveyor down" pin='A01' state='on'></div></code>
	Conveyor Left	<code><div class="conveyor left" pin='A01' state='on'></div></code>

Actuators

Actuators can be used to show radial or linear movement. There exist 3 options and they can be turned on/off;

- Rotation
- Horizontal
- Vertical

Animation	Description	Code
	Rotation	<code><div class="actuator rotation" pin='A01' state='on'></div></code>
	Horizontal	<code><div class="actuator horizontal" pin='A01' state='on'></div></code>
	Vertical	<code><div class="actuator vertical" pin='A01' state='on'></div></code>

Bulbs

Bulbs can be used to show binary information such as sensors being activated. The status is shown in two colors:

- Off - White
- On - Green

Animation	Description	Code
	Bulb Off	<code><div class="bulb" pin="A00" state="off"></div></code>
	Bulb Off	<code><div class="bulb" pin="A00" state="off"></div></code>
	Bulb Off (Resized)	<code><div class="bulb" pin="A00" state="off" style="width: 50px; height: 50px;"></div></code>
	Bulb On (Resized)	<code><div class="bulb" pin="A00" state="on" style="width: 50px; height: 50px;"></div></code>

Switches

Switches can be used for controlling hardware. They send a binary 1 or 0 (on or off) signal. The pin address is specified by the "pin" attribute. The associate hardware context.

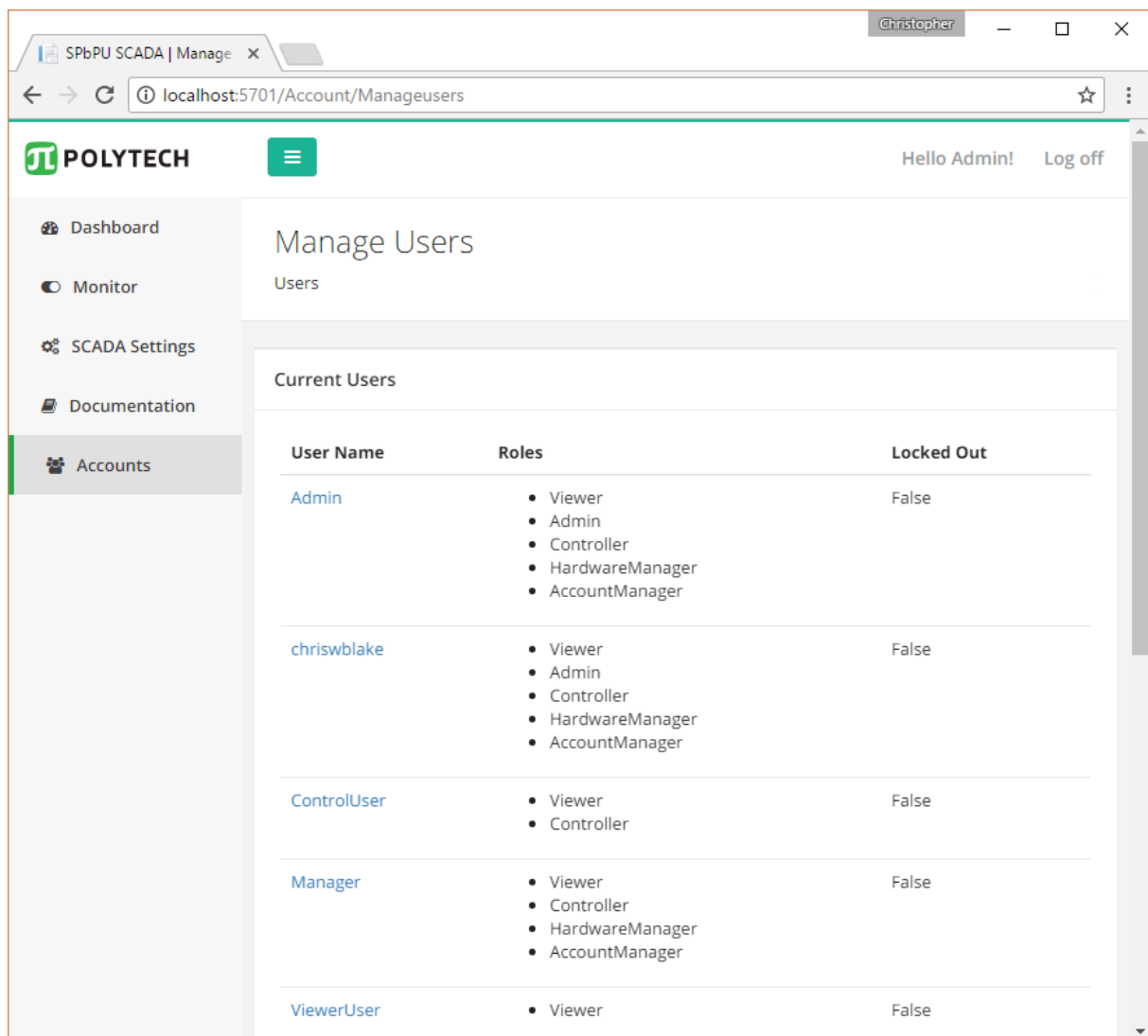
Animation	Description	Code
	Switch	<code><input type="checkbox" class="switch-onoff" pin="A00"></code>
	Switch (sized)	<code><input type="checkbox" class="switch-onoff" pin="A00" style="width: 70px; height: 40px;"></code>

Accounts Page

Accounts with the role of “Admin” or “Account Manager” may modify roles for other users. There exist five roles, shown below. When a user registers for the site, they are granted, by default, the “Viewer” role.

1. Admin – Access to all system functions.
2. Account Manager – Can modify account roles.
3. Hardware Manager – Can modify hardware for the SCADA systems.
4. Controller – Can issue commands to hardware.
5. Viewer – Can view the hardware.

Note: These roles are not implicit, which means they are fully separated. Example: If a user has the role “Controller” but not “Viewer”, then then this user would only be able to issue commands but not see the running systems. As such, attempting to issue commands would prove difficult as the only option would be to issue commands via the Hardware API.



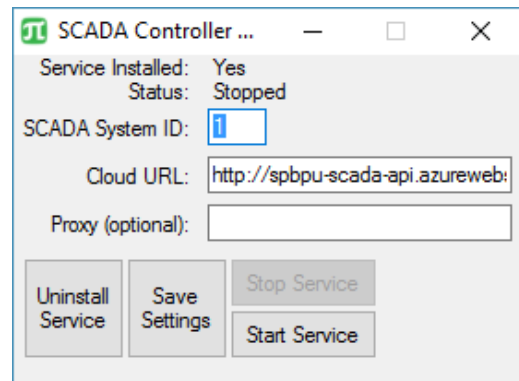
The screenshot shows a web browser window with the URL `localhost:5701/Account/Manageusers`. The page title is "Manage Users" and the user is logged in as "Admin". The left sidebar contains navigation options: Dashboard, Monitor, SCADA Settings, Documentation, and Accounts (selected). The main content area shows a table of "Current Users".

User Name	Roles	Locked Out
Admin	<ul style="list-style-type: none">ViewerAdminControllerHardwareManagerAccountManager	False
chriswblake	<ul style="list-style-type: none">ViewerAdminControllerHardwareManagerAccountManager	False
ControlUser	<ul style="list-style-type: none">ViewerController	False
Manager	<ul style="list-style-type: none">ViewerControllerHardwareManagerAccountManager	False
ViewerUser	<ul style="list-style-type: none">Viewer	False

SCADA Controller

The actual SCADA system is controlled by a Windows service, which is installed locally. This windows service connects to the local SCADA system via local LAN and communicates with the cloud service.

Upon connection with the cloud, it uses the specified SCADA System ID, to retrieve information about the local SCADA system. It uses these system details for polling the system and then sends these updates live to the cloud. If the SCADA system is reconfigured from the cloud it receives these updates automatically.



Downloading

The "SCADA Controller" installation software is available through the web interface, under the documentation area. These instructions and more are additionally available in the website's documentation area.

Installation

The service may be installed in one step. Press the "Install Service" button. Note: Administrator privileges are required. After installation, the service may be configured and started.

Prerequisites

- Windows XP SP3
- .Net Framework 4.0 or newer
- Internet Access

Interface

Field Descriptions

1. Service Installed – If the service is already installed on this system.
2. Status – Current operating status of the service.
3. SCADA System ID – The ID of this SCADA system. This can be seen in the web interface.
4. Cloud URL – The add to the web interface. (Hardware API connection is implied)
5. Proxy URL – It may be required to use a proxy service. URL and credentials may be added here. Warning: credentials are stored in plain text.

Button Descriptions

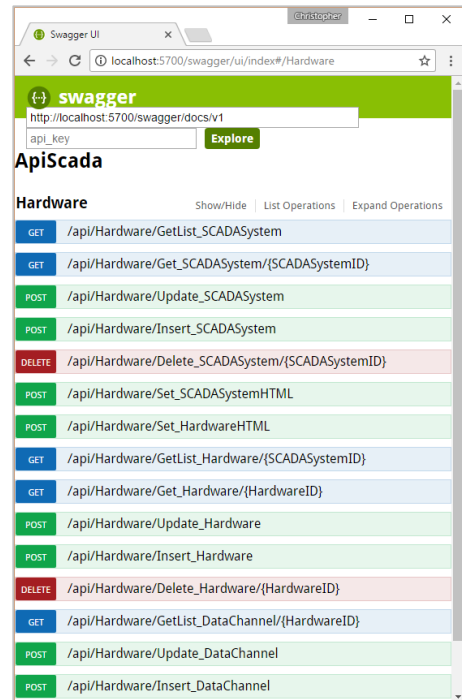
1. Install/Uninstall Service – Allow this computer to control the local SCADA system.
2. Save Settings – Saving of the SCADA System ID, cloud URL, and polling interval.
3. Stop Service – Stops the service, if it is currently running.
4. Start Service – Starts the service, if it is not running.

Hardware API

All hardware related information is controlled through a dedicated API. This API manages a separate database with four tables. These four tables are used together to describe any number of SCADA systems and record usage history.

1. SCADASystem
2. Hardware
3. DataChannel
4. Data

The actual API is with full documentation is available via the web. It is available to accounts with admin privileges.



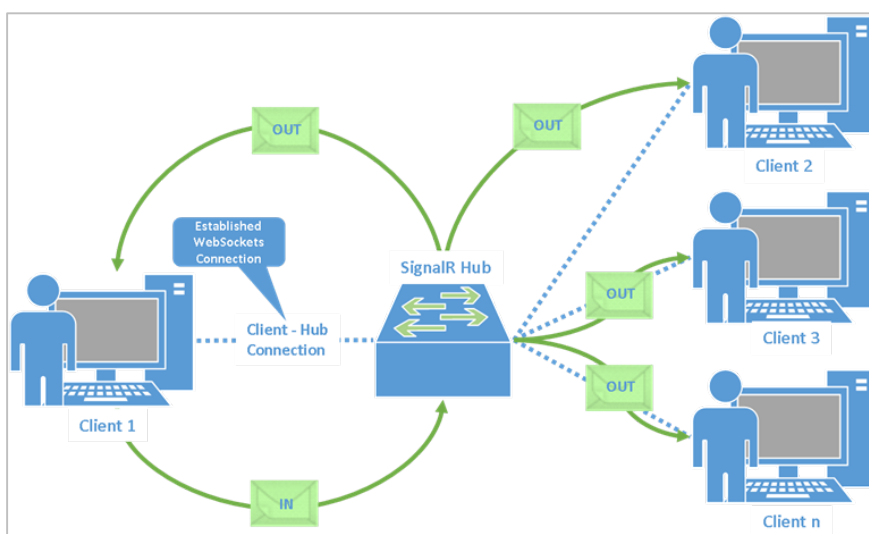
Real-time Hub

Push updates are enabled by means of a communication service known as “SignalR”. This service enables all clients such as web interfaces, SCADA Controllers, and APIs to send messages to other clients at real-time. Below is a graphic demonstrating this process.



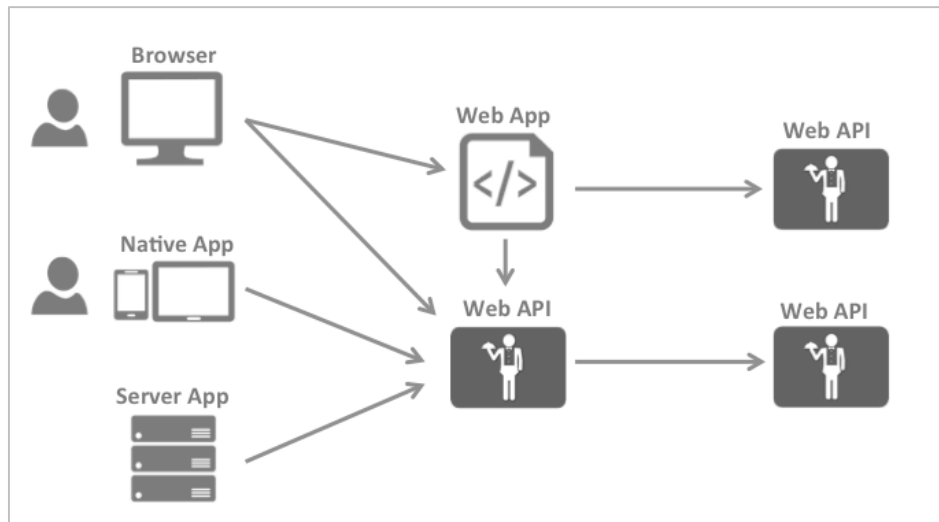
Process

1. Client connects to real-time hub and a connection ID is assigned.
2. Hub requests client details such as name and client type.
3. Implicit commands are made available on both clients and hub.
4. A client calls a method on the hub, providing parameters if required.
5. Using the client list, the appropriate response is sent to all clients.



Identity Server

Authentication to all system resources is maintained by a service known as “Identity Server 4”. This is a robust and tested service designed by security experts at Microsoft. When a user logs into the system, the identity server provides a token to this user. From this point forward, this token is paired with all future communications, authenticating the request. Below is a graphic from the Identity Server 4 website demonstrating this concept.



Future Considerations

The service is intended to be an ongoing development, continuously adding new capabilities. Below is a list of ideas for this initiative.

- Data Mining of usage history.
 - Situation Detection – Failure prediction.
 - Process Optimization – Detection of bottlenecks.
- Mobile Applications
 - Subscribe to alerts
- Additional non-SCADA hardware services

Conclusion

Using the fundamentals and proof-of-concept work from an existing C# Windows Forms Application, a web based service has been produced for SCADA systems. to allow management and control of SCADA systems.

Utilizing five service components, a SCADA system can be configured and controlled by an authenticated user. A website provides desktop- and mobile-friendly interface for managing the SCADA hardware. An API structure is utilized to support modular functionality. A real-time hub is created for providing live updates and push notifications. A windows service connects local hardware to the cloud service. And finally, everything is secured using token-based authentication maintained by an identity server.