NETWORK AND COMMUNICATIONS MANAGEMENT
SENIOR PROJECT
NETW 497

GLOBAL SYSTEM MANAGEMENT, LLC

The Raspitor Project

The cost effective and reliable solution to

Your network physical environment variations…

DeVry University

PROFESSOR: RUBEN ARIAS

PREPARED BY THE MEMBERS OF TRISTAR INTERNATIONAL, LLC

OMAR RAHAMAN ANDY LAU ELIGIO ESPINOZA

Table of Contents

Executive Summary

The Management of Global Telecommunication Management have been apprehensive about the evolution of their company and have decided to take a more comprehensive approach to solve their problem, which will help to expand their services. The concerns include environmental conditions in their equipment room: insufficient airflow, extreme temperatures, and high humidity. The network system administrator and network operators needed warning to avoid damage to equipment and loss of data with the consequent downtime operation.

To respond to this existing apprehension, Global Telecommunication Management has called TriStar International to deliver a more contemporary network monitoring system that would meet the needs of the company. TriStar International will implement their signature network monitoring system, Raspitor that will not only boost the services offered by Global Telecommunication Management, but one that will also deliver a more consistent and reliable protection to the equipment room. The Raspitor network monitoring system is not just a networking monitoring system, but rather a compound of sophisticated pieces of equipment. The system has the ability to trigger countermeasures to minimize possible damage to the equipment. It also sends notifications via text messaging, as well as, displaying audio-visual alarms while help arrives to fix HVAC failures.

Introduction

With the existing setup and network configuration understood at Global Telecommunication Management; we at TriStar International pursue to deliver networking services that would sustain the existing and forthcoming evolution of Global Telecommunication Management. TriStar international will customize a state of the art network monitoring solution that will provide peace of mind to the corporation's day-to-day processes and improve their service reliability. TriStar International proposed solution is to implement its signature network monitoring system that would replace their existing ineffective system that functions more as a network probe than a network monitoring system. This solution would incorporate that latest hardware and software to create the most advanced environmental monitoring system available in today's technological market. This cost effective solution guarantees to escalate the network health of Global Telecommunication Management and provide the needed reliability - with minimum downtime and through reduced response-to-resolution times.

TriStar International

Over the past 10 years, TriStar International has distinguished as a leader in the implementation of environmental network monitoring systems supporting domestic and foreign organizations in 35 states and 10 countries around the world. TriStar International headquarters is located in Plantation FL, working with numerous partners as they constantly participate in the global market for state of the art IT technology. The project team of TriStar International has generated routine resolutions for its partners.

Our Strengths

- Encompass a team of IT experts, TriStar International emphasize on generating resolutions for all methodological processes, while exercising pride in producing the best economical results in today’s market.

Our Core values

- To deliver superior service. In our efficient labs we have designed, developed, and tested customized solutions for network environment monitoring with swift response countermeasures.
- Our network engineers develop the most advance methods to create a sophisticated monitoring system using the latest software and hardware available in the market as well as, customization to specific company’s requirements.
- We are Avant guard in hardware and software integration and deliver nothing but the best from the top manufacturers in the technological world.

TriStar International Staff

Plantation Headquarters

- Our headquarters in Plantation accommodates 50 fulltime employees.
- Our headquarters in Plantation employs 20 certified senior network systems.

UK Manufacturing Company

- Our branch office in the UK accommodates 45 fulltime employees.
- Our branch office in the UK comprises of a Sales team, an Engineering division with 12 certified electronic engineers; as well as 10 certified software developers and programmers.

Global Telecommunication Management Organizational Structure

Global Telecommunications Management was established in 2005 to provide low cost telecommunications services to companies and home users around the world; especially to Europe and South America. Through the years of continued success, they have decided to increase their operation to areas in Security and Information Technology in the locations of Delaware and Baltimore. These locations are divided in four departments: Executives, Marketing and Finance, Sales Support and Customer Service. Currently, Global Telecommunications Management consists of 250 with a projected increase in the next 5 years to incorporate 60 more employees in the operation and engineering departments.

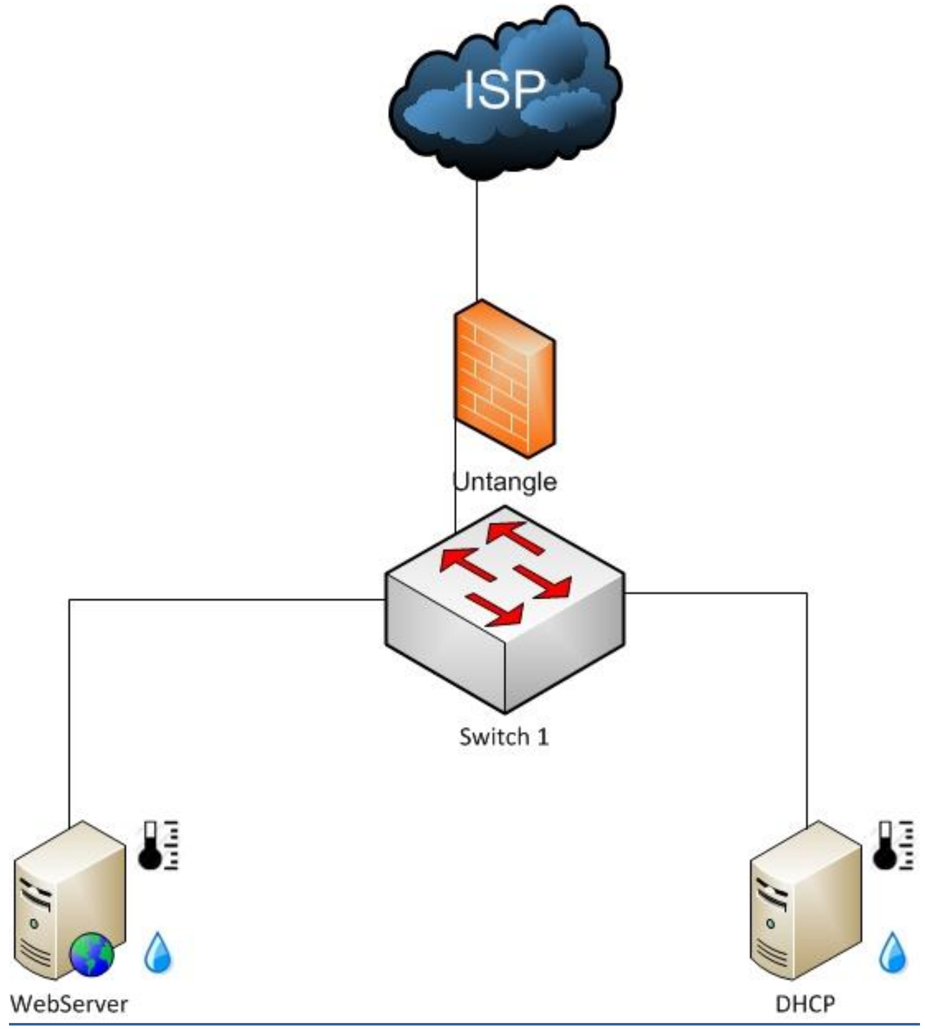
Summary of Existing Network Structure

After performing a site survey TriStar International provided a comprehensive assessment of the Global System Management’s situation. TriStar determined that the quality of service provided by their current network monitoring system was of inferior quality and did not provide the advantages of our palliative countermeasures. They relied on software that only monitored internal temperature of the servers disregarding environmental factors as temperature and humidity. We detected insufficient airflow, extreme temperature, and high humidity for an equipment room.

Consequently, we proposed the implementation of Raspitor monitoring system, our signature customizable environment monitoring system for equipment rooms.

The next diagram shows their present network infrastructure, which lacks of temperature and humidity sensors, as well as the countermeasures TriStar offers in their solution.

GTM Existing Network Diagram



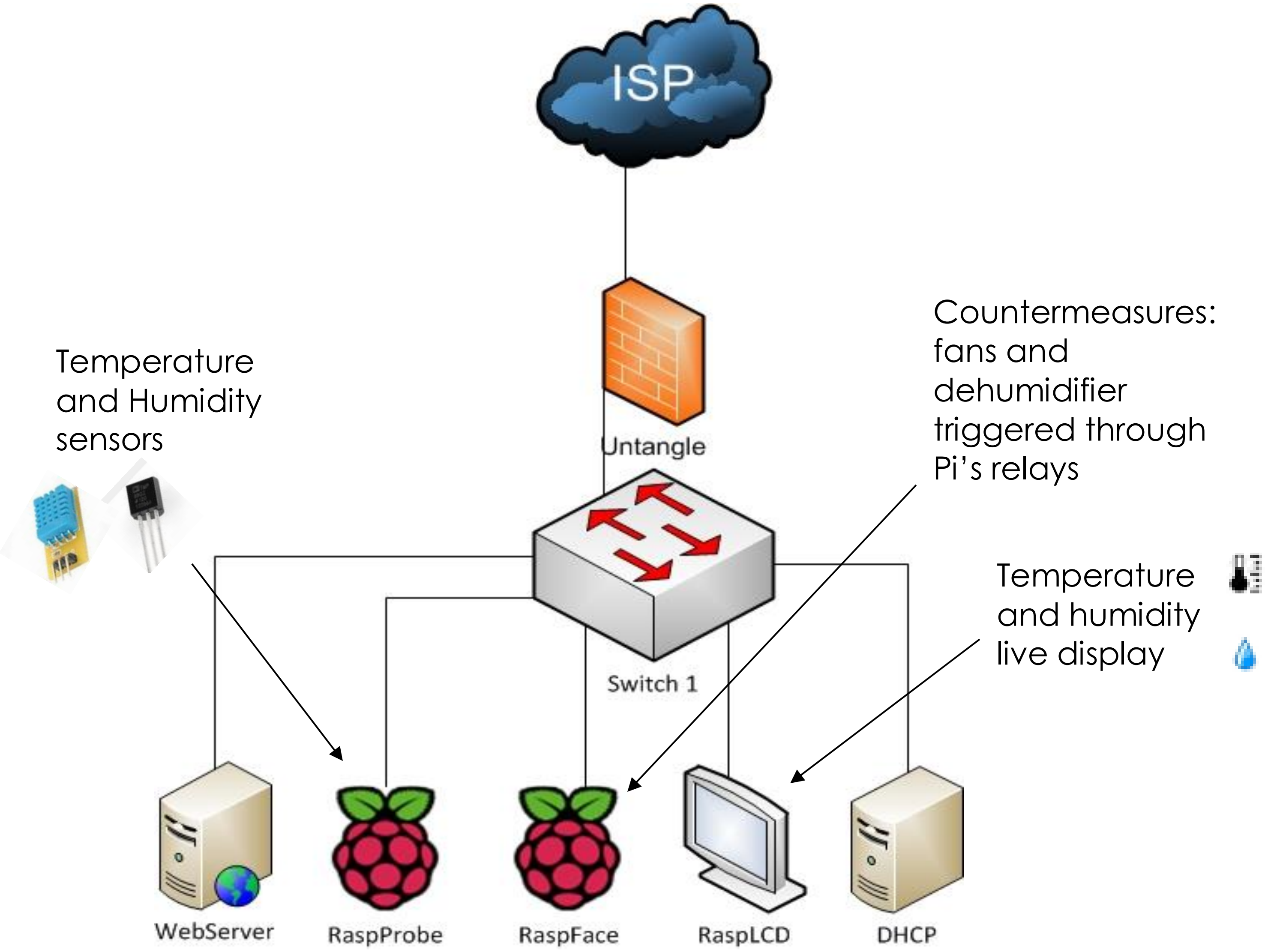
Summary of Proposed Network Structure

The proposed network structure will consist of the existing network topology that Global Telecommunication Management has with the addition of the Raspitor monitoring system and the countermeasures system proposed by TriStar International.

The proposed network structure consist of the RaspProbe server, that will gather temperature and humidity readings; the RaspFace server, which will control and trigger the countermeasure mechanisms; and the RaspLCD which will display temperature and humidity readings in real-time.

The next diagram depicts the proposed network monitoring solution with brief notes on the performance of main components of our signature network monitoring product.

TRISTAR Proposed Network Diagram



Temperature and Humidity sensors

Countermeasures: fans and dehumidifier triggered through Pi’s relays

Temperature and humidity live display

Technical Goals and Constraints

Technical Goal	Level of Importance	Comments
Performance	Critical	To provide optimum functioning with real response time to +/- 3 seconds after processing detection of variations in temperature and humidity, and an accuracy of +/- 0.5 °C and +/- 5%RH
Availability	Important	The network design process should have a 99% availability, which translate into no more than 3.65 Days of downtime a year (87.6 hours)
Adaptability to new technology	Critical	It is imperative that the network monitoring system include features that allows it to be functional in any working environment
Scalability	Critical	It is vital that the network monitoring system be scalable to suit the needs and wants. Temperature sensor response time obeys to amount of sensors per RaspProbe
Manageability	Important	The proposed network monitoring system must be clear and concise, so it is accessible to administrator and authorized users
Upgrade	Important	Devices that are outdated must be upgraded to support the current application or feature and according to the limitations of mini-micro-card

Benefits of solution to Global System Management, LLC

After installing Raspitor Monitoring System GSM will be able to prevent damage to their equipment due to unnoticed HVAC failures. Raspitor early warning system will alert network administrators of variations in temperature and humidity levels, via local audio-visual alarms and email notification. At the same time, it will trigger countermeasures that use powerful fans and extractors to avoid damage to the equipment and downtime or shutdown of service to GMS clients.

In addition, network administrators will be able to watch the server room from anywhere through the website and look at temperature and humidity levels history, as well as comparison between sensors.

Raspitor Monitoring System will enable GSM to reduce its annual equipment replacement budget, and network "up" time will be higher. It will also cut response time, and IT personnel will be more efficient.

Main Benefits:

- Alert network administrator and network operators of HVAC failures, which will prevent damage to computer equipment valued in thousands of dollars.
- Increases network availability and employee productivity.
- Reduced response-to-resolution time

Description of Hardware used

Raspberry Pi:

The Raspberry Pi is a single-board computer developed in the UK by the Raspberry Pi Foundation. The Raspberry Pi is a credit-card sized computer that plugs into your TV and a keyboard. It's a capable little PC which can be used for many of the things that your desktop PC does, like spreadsheets, word-processing, and games, as well as plays high-definition video. The design is based around a Broadcom BCM2835 SoC, which includes an ARM1176JZF-S 700 MHz processor, Video Core IV GPU, and 512 Megabytes of RAM. This revision 2.0 board features two mounting holes for easy installation; a built-in reset circuit, and can be powered via the USB data ports. The design does not include a built-in hard disk or solid-state drive, instead relying on an SD card (not included) for booting and long-term storage. The Raspberry Pi is intended to run Linux kernel based operating systems. Component colors may vary.

Raspberry Pi-Face:

The Raspberry Pi-Face manufactured by PIFACE is an add on daughter board that allows the raspberry PI to control 2 relays, 8 open collector outputs, and 4 tactile switches. The Raspberry Pi Face allows users to connect external objects into the Raspberry Pi and have it communicate with the device. Because the Pi-Face contains the relays and the open-collectors; It allows the Raspitor to control external devices by switching the devices on or off. Programmed with Python, the Pi-Face can be customized to the needs and requirements of the User to work with their project.

DHT 11 Humidity Sensor:

The DHT Humidity Sensors allows the Raspitor to detect changes in the humidity. The resolution of DHT11 is 8 bit allowing the device to convert analog signals to digital signals in 256 different stages. The Humidity accuracy range for temperature is between 0-50°C (± 2°C). Secondly, the humidity range for the DHT sensors is 20-80% Relative humidity (±5% RH)

DS18B20 Temperature sensor:

The DS18B20 Temperature sensors allows the Raspitor to detect temperature with precise accuracy. Produced by Maxim IC, the Resolution of the DS18B20 can range between 9bits to 12 Bits. The temperature sensing range of the DS18B20 is -55C to 125C (±0.5C). This high accuracy allows the device to precisely measure the temperature of the environment. The communication uses a 1 wire interface thus allowing multiple sensors to the configured in parallel and have the sensors communicate to the Raspberry Pi using only 1 data line.

Description of Software used

Apache

The Apache HTTP server project is an initiative for communities to develop and maintain an open-source HTTP server platform for UNIX and Windows. The goal is to provide a secure, efficient and extendible HTTP interface for current HTTP standards. The Raspitor web server utilizes apache as a service for hosting our Raspitor web interface.

MySQL

MySQL is an open source application used for providing users access to numerous database spanning across several networks. Because of its open source, MySQL has been a popular choice for database for use in web applications and is widely used open source projects that require a database as its core of the project. The Raspitor utilize the database heavily for environmental monitoring by storing data such as time, sensor identification, temperature, and humidity. By having a MySQL database, it frees up the limited resources that the Raspberry Pi can supply.

pChart2

PChart 2 is a software library released under the GNU GPLv3 license intended for open source development and academic use. Many organizations and companies such as NASA utilize PChart because of its functionality and customizations. The Raspitor web server heavily depends on the PChart library for their environmental monitoring interface. The main keys on PChart 2 include:

- Open Source

- Object Oriented Interface
- Auto Scaling and Units
- Supports line graphs, pie charts, bar graphs, and zone graphs
- Local storage for images

Python:

Python is a programming language that allows you to integrate system effectively. The Python implementation is under the open source license allowing it to be freely used, distributed and for commercial use. Python is cross platform compatible allowing developers to move between platforms seamlessly. Because Python encompass many tasks it can be used for a variety of function from executing commands to processing information. The Raspitor depends heavily on the Python programming platform for their customized software. Their ease of use allows any beginning programmer to start interacting with the Raspberry Pi from the start.

Python-MySQL dB:

MySQL dB is an API that allows programmers to provide an interface to a MySQL database through the language of Python. The software runs under the GNU GPLv3 License for software development to create custom script through the use of Python. The Raspitor project depends heavily on the Python-MySQL dB for storing, gathering, and editing data from the MySQL database hosted on the Raspitor database.

Linux:

Linux is an open source operating system used by many users and organizations because of its open source development. Many communities have contributed into the operating system improving its functionality and performance. The core concept of the Raspitor is to create a low-cost open source environmental monitoring system. The system runs many variants of Linux each with a purpose in the network. The Raspberry Pi runs Raspbian, a variant of Debian which was exclusively created for the Raspberry Pi platform. The Web server and DHCP server runs CENTOS a community dedicated in creating an operating system used in enterprise network.

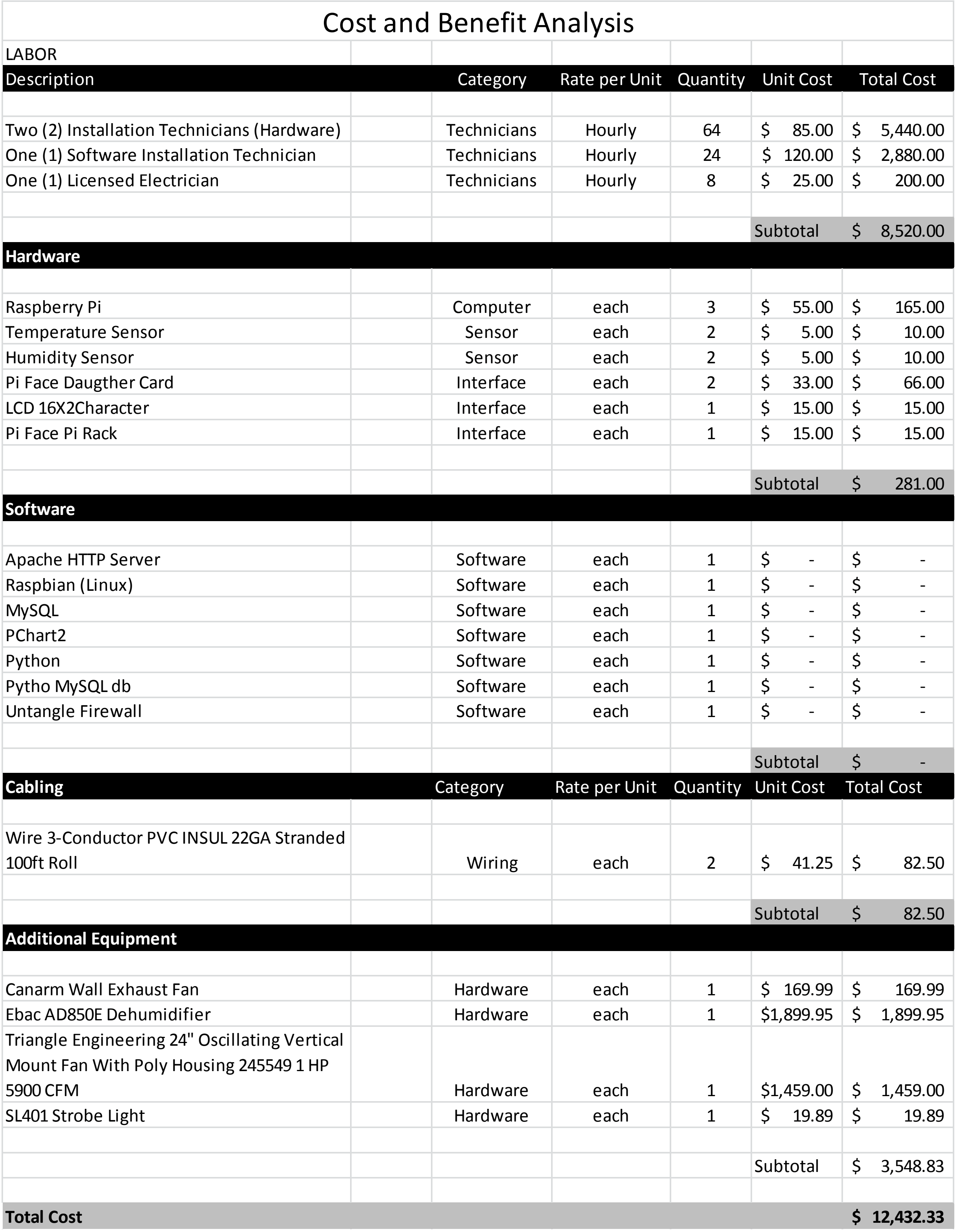
Untangle Firewall:

The Untangle Firewall is an open source Linux based multifunctional firewall which includes most of the applications that we currently find in many network gateway today.

The firewall is single software that can be charged with as many applications and control that the company wants to implement. In the Global Management System, we have included the following applications:

- *Bandwidth Control:* this application is used to manage and observe bandwidth used in the network and also to arrange time critical applications.
- *Intrusion Prevention:* this is an invasion discovery system that will capture all traffic and identifies malicious actions, this uses signature recognition; a process that draws upon a database or known attack pattern.
- *Spam Blocker:* this application is capable of detecting unwanted bulk mail sent from an image server or from a legitimate email address.

Cost and Benefit Analysis

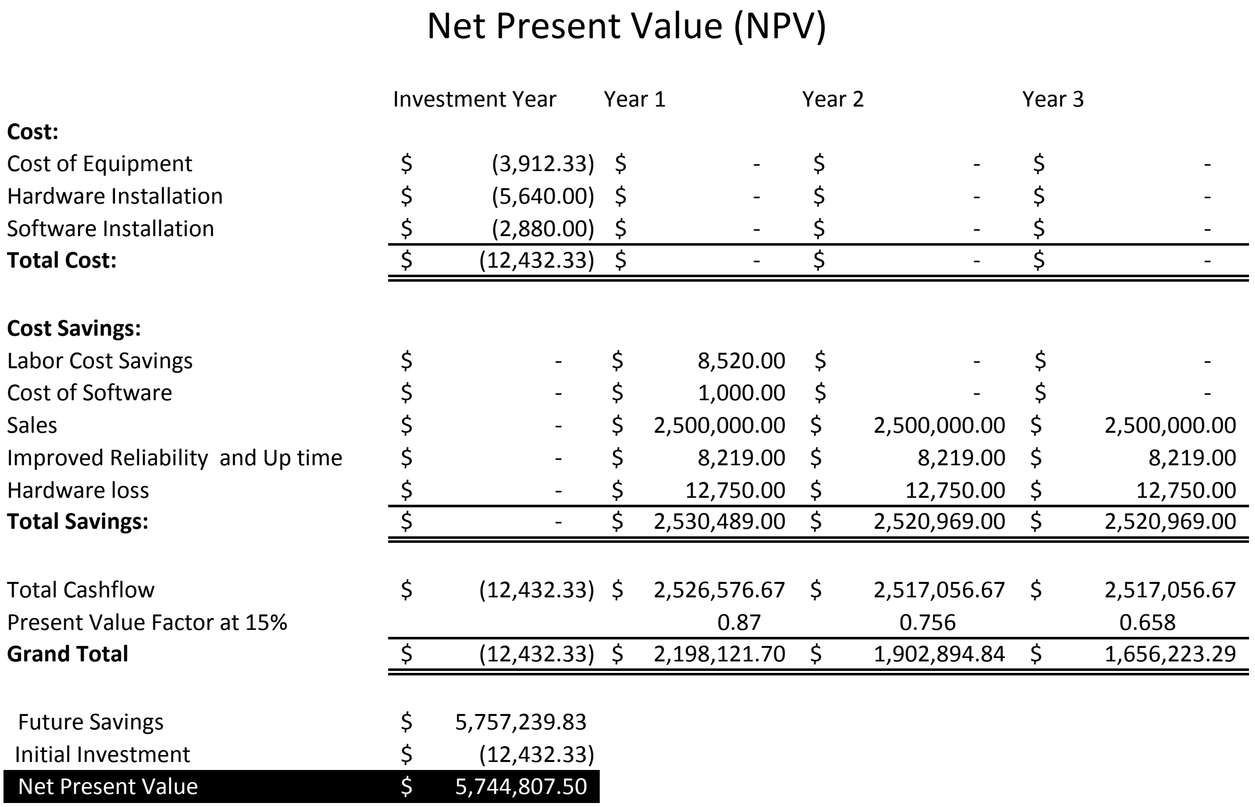


Cost and Benefit Analysis

LABOR

Description	Category	Rate per Unit	Quantity	Unit Cost	Total Cost
Two (2) Installation Technicians (Hardware)	Technicians	Hourly	64	$ 85.00	$ 5,440.00
One (1) Software Installation Technician	Technicians	Hourly	24	$ 120.00	$ 2,880.00
One (1) Licensed Electrician	Technicians	Hourly	8	$ 25.00	$ 200.00
				Subtotal	$ 8,520.00
Hardware					
Raspberry Pi	Computer	each	3	$ 55.00	$ 165.00
Temperature Sensor	Sensor	each	2	$ 5.00	$ 10.00
Humidity Sensor	Sensor	each	2	$ 5.00	$ 10.00
Pi Face Daugther Card	Interface	each	2	$ 33.00	$ 66.00
LCD 16X2Character	Interface	each	1	$ 15.00	$ 15.00
Pi Face Pi Rack	Interface	each	1	$ 15.00	$ 15.00
				Subtotal	$ 281.00
Software					
Apache HTTP Server	Software	each	1	$ -	$ -
Raspbian (Linux)	Software	each	1	$ -	$ -
MySQL	Software	each	1	$ -	$ -
PChart2	Software	each	1	$ -	$ -
Python	Software	each	1	$ -	$ -
Pytho MySQL db	Software	each	1	$ -	$ -
Untangle Firewall	Software	each	1	$ -	$ -
				Subtotal	$ -
Cabling	Category	Rate per Unit	Quantity	Unit Cost	Total Cost
Wire 3-Conductor PVC INSUL 22GA Stranded 100ft Roll	Wiring	each	2	$ 41.25	$ 82.50
				Subtotal	$ 82.50
Additional Equipment					
Canarm Wall Exhaust Fan	Hardware	each	1	$ 169.99	$ 169.99
Ebac AD850E Dehumidifier	Hardware	each	1	$1,899.95	$ 1,899.95
Triangle Engineering 24" Oscillating Vertical Mount Fan With Poly Housing 245549 1 HP 5900 CFM	Hardware	each	1	$1,459.00	$ 1,459.00
SL401 Strobe Light	Hardware	each	1	$ 19.89	$ 19.89
				Subtotal	$ 3,548.83
Total Cost					**$ 12,432.33**

Net Present Value (NPV) over a period of Three Years



Net Present Value (NPV)

	Investment Year	Year 1	Year 2	Year 3
Cost:				
Cost of Equipment	$ (3,912.33)	$ -	$ -	$ -
Hardware Installation	$ (5,640.00)	$ -	$ -	$ -
Software Installation	$ (2,880.00)	$ -	$ -	$ -
Total Cost:	$ (12,432.33)	$ -	$ -	$ -
Cost Savings:				
Labor Cost Savings	$ -	$ 8,520.00	$ -	$ -
Cost of Software	$ -	$ 1,000.00	$ -	$ -
Sales	$ -	$ 2,500,000.00	$ 2,500,000.00	$ 2,500,000.00
Improved Reliability and Up time	$ -	$ 8,219.00	$ 8,219.00	$ 8,219.00
Hardware loss	$ -	$ 12,750.00	$ 12,750.00	$ 12,750.00
Total Savings:	$ -	$ 2,530,489.00	$ 2,520,969.00	$ 2,520,969.00
Total Cashflow	$ (12,432.33)	$ 2,526,576.67	$ 2,517,056.67	$ 2,517,056.67
Present Value Factor at 15%		0.87	0.756	0.658
Grand Total	$ (12,432.33)	$ 2,198,121.70	$ 1,902,894.84	$ 1,656,223.29
Future Savings	$ 5,757,239.83			
Initial Investment	$ (12,432.33)			
Net Present Value	$ 5,744,807.50			

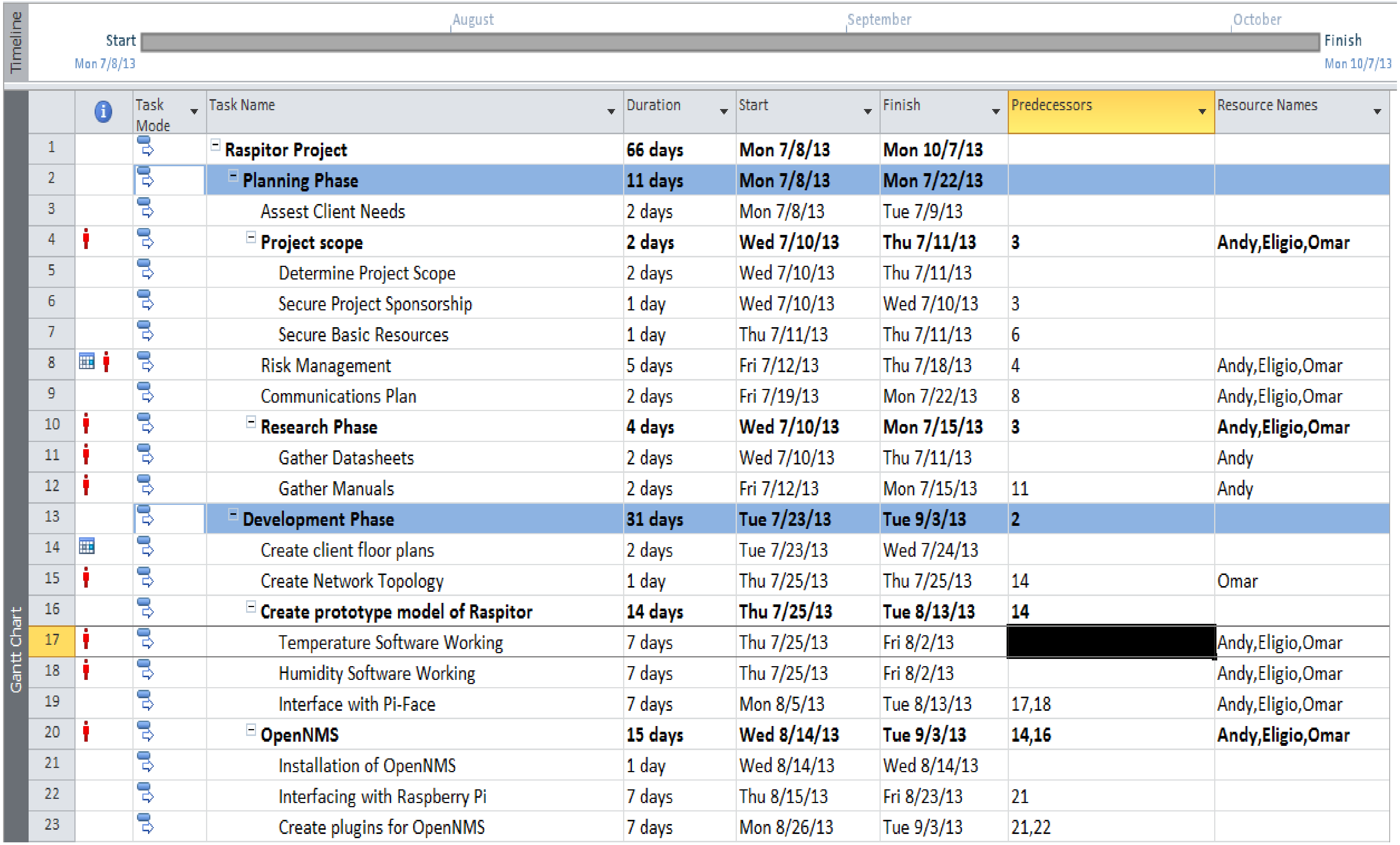
As we can observe in the NPV analysis, the cost of the investment is infinitesimal $12,432.33 in comparison to the gigantic sales and future savings of $5,757,239.83 at the end of the third year, which results in an NPV of $5,744,807.50 at PVF of 15% according to the actual value table.

We chose NPV method to calculate if the investment was worth the value for the organization. Even though this method estimates future cash flows which may be far from actual results, it is a more reliable method to appraise an investment. The advantage of NPV is that it accounts for the time value of money (the value of a dollar to be received in the future is less than the value of a dollar on hand today) in contrast to

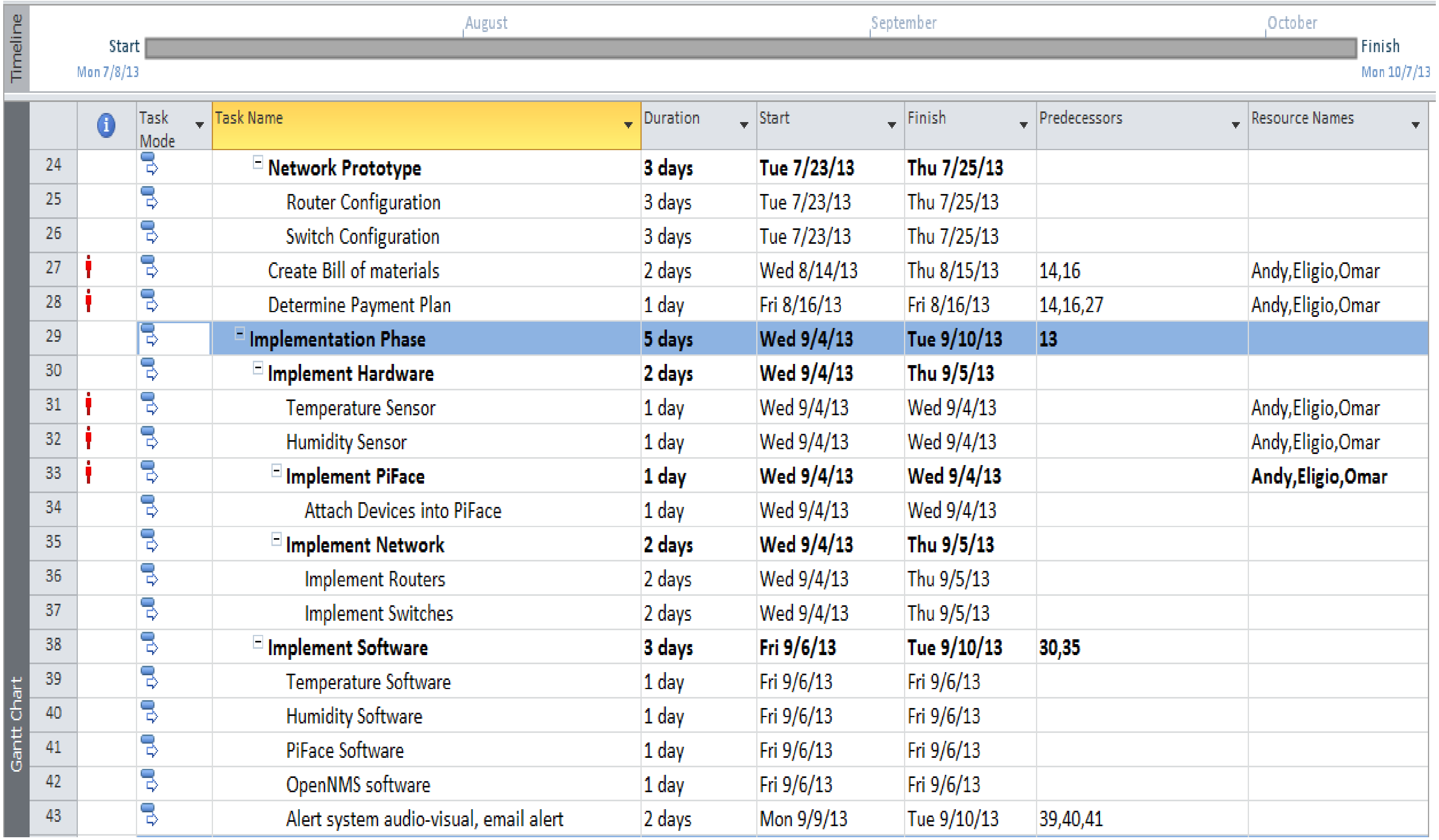
Pay Back Period and Accounting Rate of Return, which do not discount future cash flows.

It is quite obvious that the investment is a terrific choice even though it is a system prototype that it has proved to work with huge savings for the company and minimum down time with a NPV greater than zero.

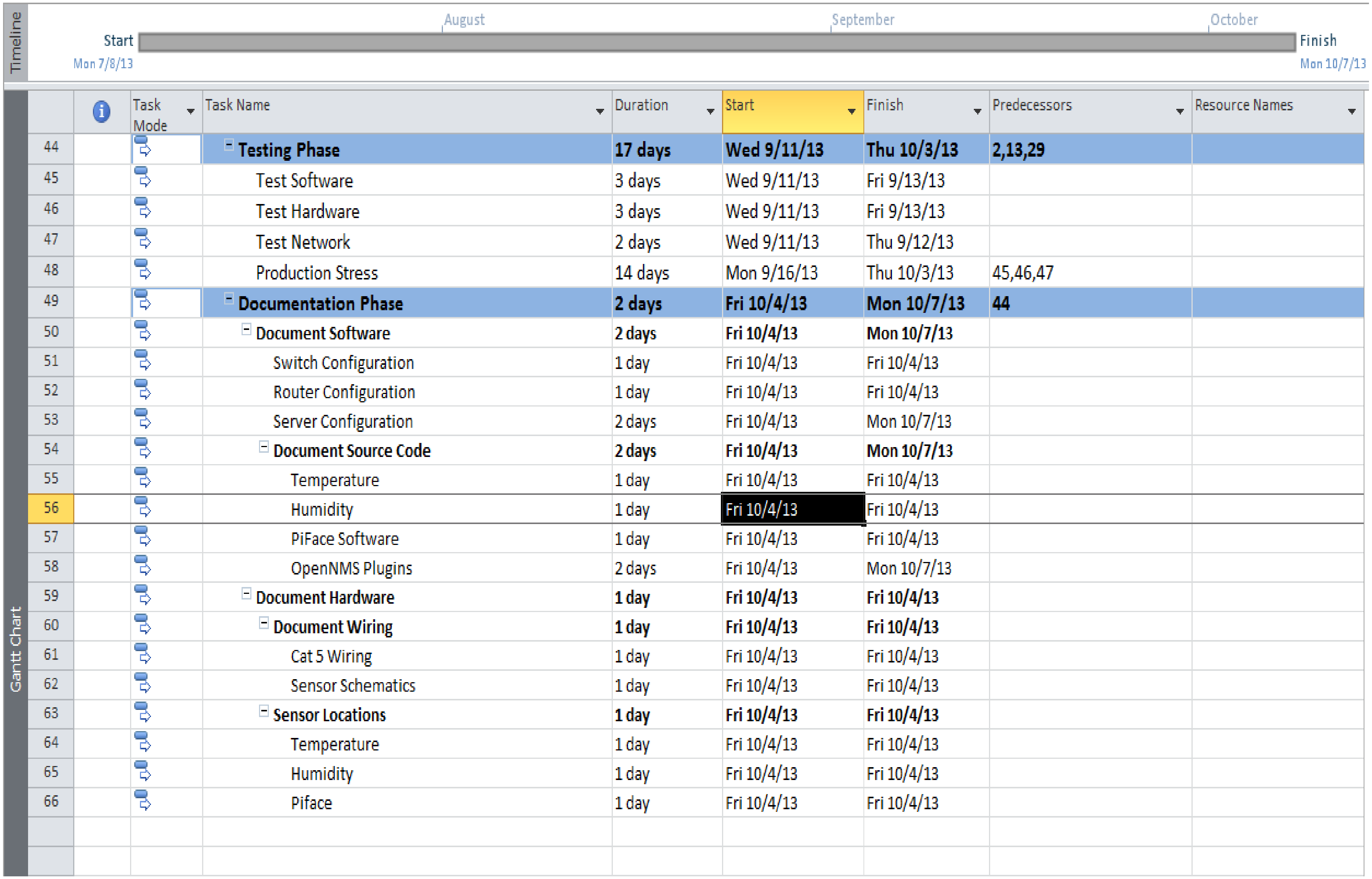
Gantt Chart



	ⓘ	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names
1		⇨	**Raspitor Project**	**66 days**	**Mon 7/8/13**	**Mon 10/7/13**		
2		⇨	**Planning Phase**	**11 days**	**Mon 7/8/13**	**Mon 7/22/13**		
3		⇨	Assest Client Needs	2 days	Mon 7/8/13	Tue 7/9/13		
4	👤	⇨	**Project scope**	**2 days**	**Wed 7/10/13**	**Thu 7/11/13**	**3**	**Andy,Eligio,Omar**
5		⇨	Determine Project Scope	2 days	Wed 7/10/13	Thu 7/11/13		
6		⇨	Secure Project Sponsorship	1 day	Wed 7/10/13	Wed 7/10/13	3	
7		⇨	Secure Basic Resources	1 day	Thu 7/11/13	Thu 7/11/13	6	
8	▦ 👤	⇨	Risk Management	5 days	Fri 7/12/13	Thu 7/18/13	4	Andy,Eligio,Omar
9		⇨	Communications Plan	2 days	Fri 7/19/13	Mon 7/22/13	8	Andy,Eligio,Omar
10	👤	⇨	**Research Phase**	**4 days**	**Wed 7/10/13**	**Mon 7/15/13**	**3**	**Andy,Eligio,Omar**
11	👤	⇨	Gather Datasheets	2 days	Wed 7/10/13	Thu 7/11/13		Andy
12	👤	⇨	Gather Manuals	2 days	Fri 7/12/13	Mon 7/15/13	11	Andy
13		⇨	**Development Phase**	**31 days**	**Tue 7/23/13**	**Tue 9/3/13**	**2**	
14	▦	⇨	Create client floor plans	2 days	Tue 7/23/13	Wed 7/24/13		
15	👤	⇨	Create Network Topology	1 day	Thu 7/25/13	Thu 7/25/13	14	Omar
16		⇨	**Create prototype model of Raspitor**	**14 days**	**Thu 7/25/13**	**Tue 8/13/13**	**14**	
17	👤	⇨	Temperature Software Working	7 days	Thu 7/25/13	Fri 8/2/13	[redacted]	Andy,Eligio,Omar
18	👤	⇨	Humidity Software Working	7 days	Thu 7/25/13	Fri 8/2/13		Andy,Eligio,Omar
19		⇨	Interface with Pi-Face	7 days	Mon 8/5/13	Tue 8/13/13	17,18	Andy,Eligio,Omar
20	👤	⇨	**OpenNMS**	**15 days**	**Wed 8/14/13**	**Tue 9/3/13**	**14,16**	**Andy,Eligio,Omar**
21		⇨	Installation of OpenNMS	1 day	Wed 8/14/13	Wed 8/14/13		
22		⇨	Interfacing with Raspberry Pi	7 days	Thu 8/15/13	Fri 8/23/13	21	
23		⇨	Create plugins for OpenNMS	7 days	Mon 8/26/13	Tue 9/3/13	21,22	

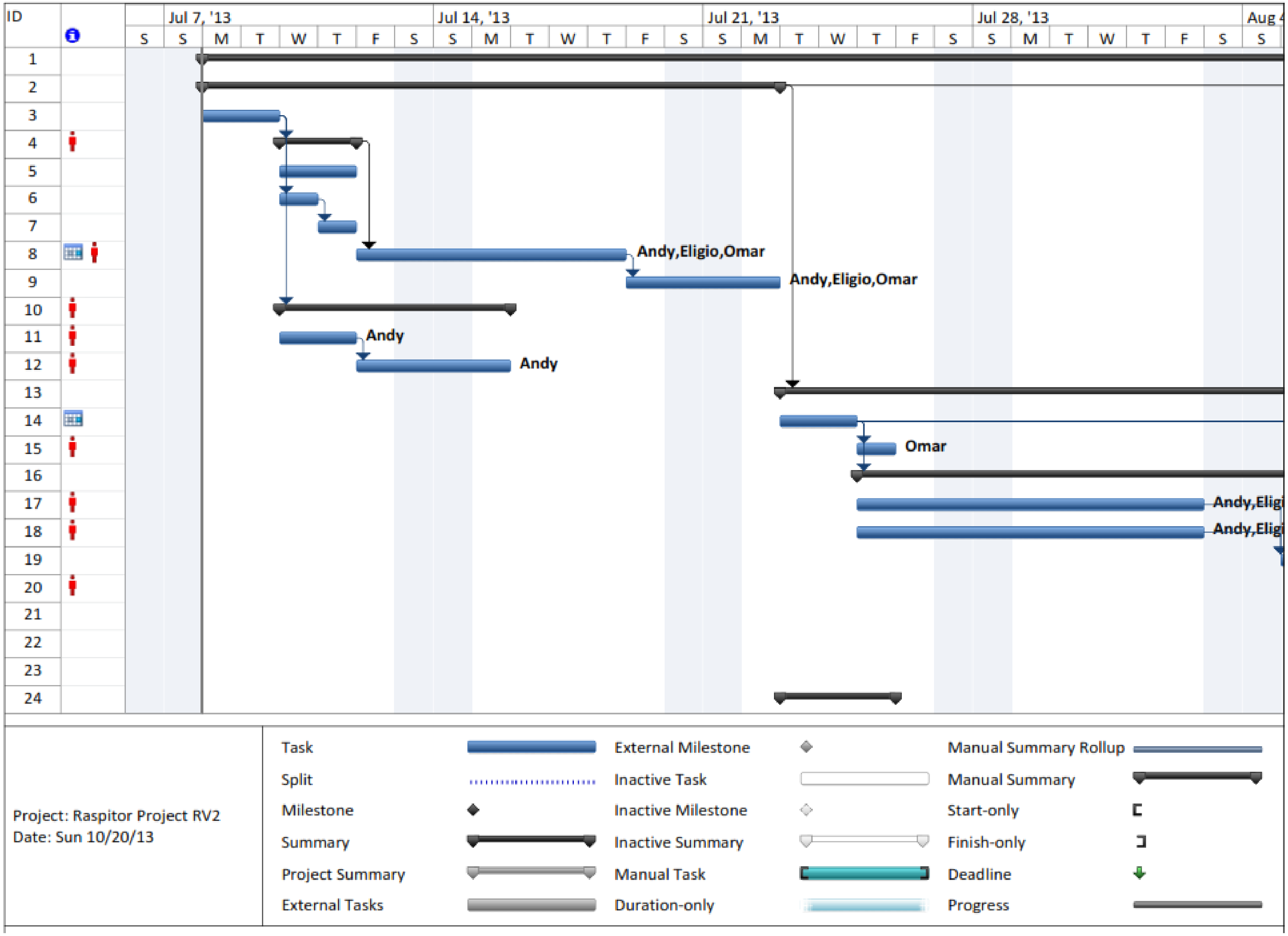


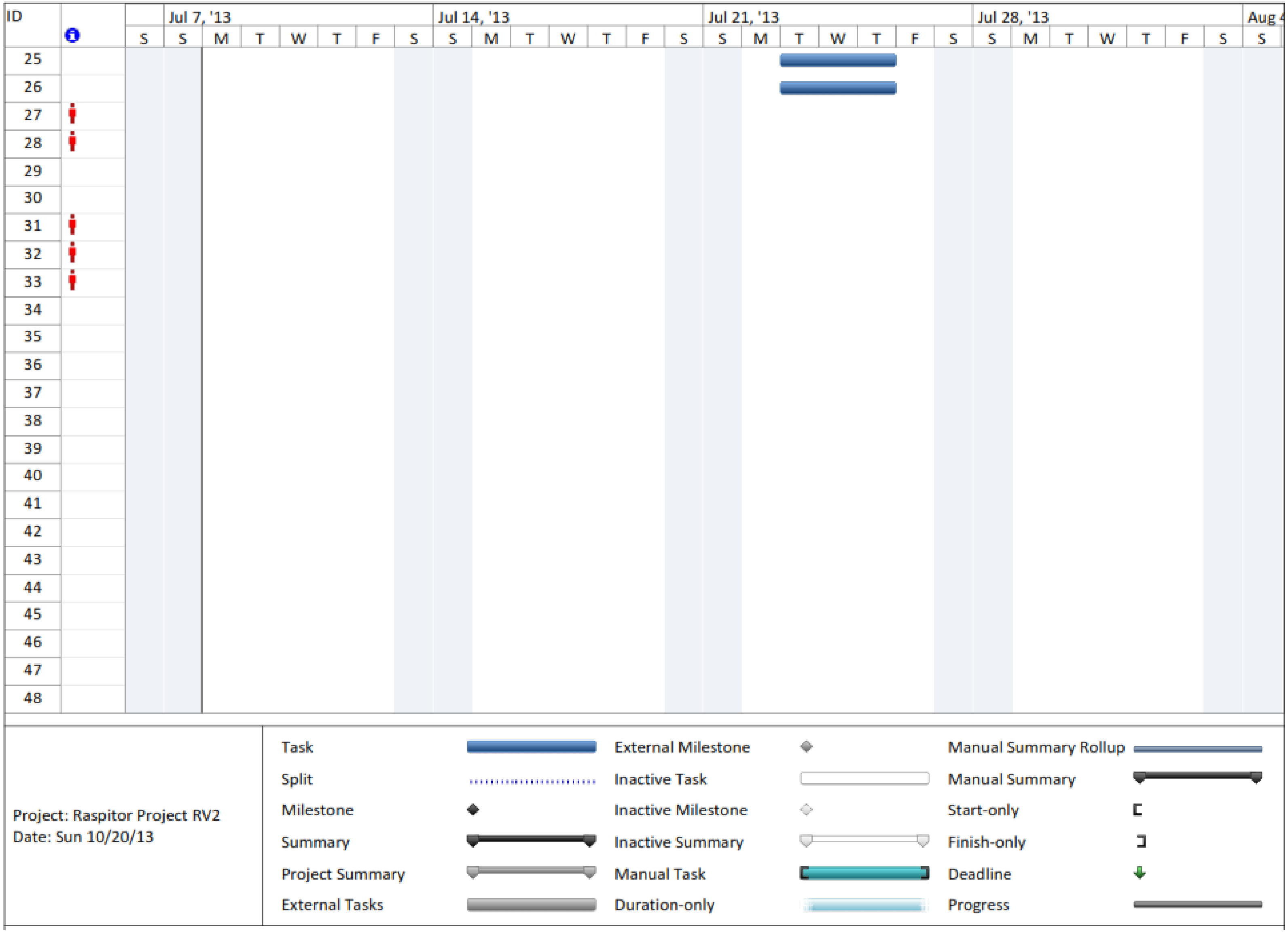
	ℹ	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names
24		⇨	**Network Prototype**	**3 days**	**Tue 7/23/13**	**Thu 7/25/13**		
25		⇨	Router Configuration	3 days	Tue 7/23/13	Thu 7/25/13		
26		⇨	Switch Configuration	3 days	Tue 7/23/13	Thu 7/25/13		
27	🚹	⇨	Create Bill of materials	2 days	Wed 8/14/13	Thu 8/15/13	14,16	Andy,Eligio,Omar
28	🚹	⇨	Determine Payment Plan	1 day	Fri 8/16/13	Fri 8/16/13	14,16,27	Andy,Eligio,Omar
29		⇨	**Implementation Phase**	**5 days**	**Wed 9/4/13**	**Tue 9/10/13**	**13**	
30		⇨	**Implement Hardware**	**2 days**	**Wed 9/4/13**	**Thu 9/5/13**		
31	🚹	⇨	Temperature Sensor	1 day	Wed 9/4/13	Wed 9/4/13		Andy,Eligio,Omar
32	🚹	⇨	Humidity Sensor	1 day	Wed 9/4/13	Wed 9/4/13		Andy,Eligio,Omar
33	🚹	⇨	**Implement PiFace**	**1 day**	**Wed 9/4/13**	**Wed 9/4/13**		**Andy,Eligio,Omar**
34		⇨	Attach Devices into PiFace	1 day	Wed 9/4/13	Wed 9/4/13		
35		⇨	**Implement Network**	**2 days**	**Wed 9/4/13**	**Thu 9/5/13**		
36		⇨	Implement Routers	2 days	Wed 9/4/13	Thu 9/5/13		
37		⇨	Implement Switches	2 days	Wed 9/4/13	Thu 9/5/13		
38		⇨	**Implement Software**	**3 days**	**Fri 9/6/13**	**Tue 9/10/13**	**30,35**	
39		⇨	Temperature Software	1 day	Fri 9/6/13	Fri 9/6/13		
40		⇨	Humidity Software	1 day	Fri 9/6/13	Fri 9/6/13		
41		⇨	PiFace Software	1 day	Fri 9/6/13	Fri 9/6/13		
42		⇨	OpenNMS software	1 day	Fri 9/6/13	Fri 9/6/13		
43		⇨	Alert system audio-visual, email alert	2 days	Mon 9/9/13	Tue 9/10/13	39,40,41	



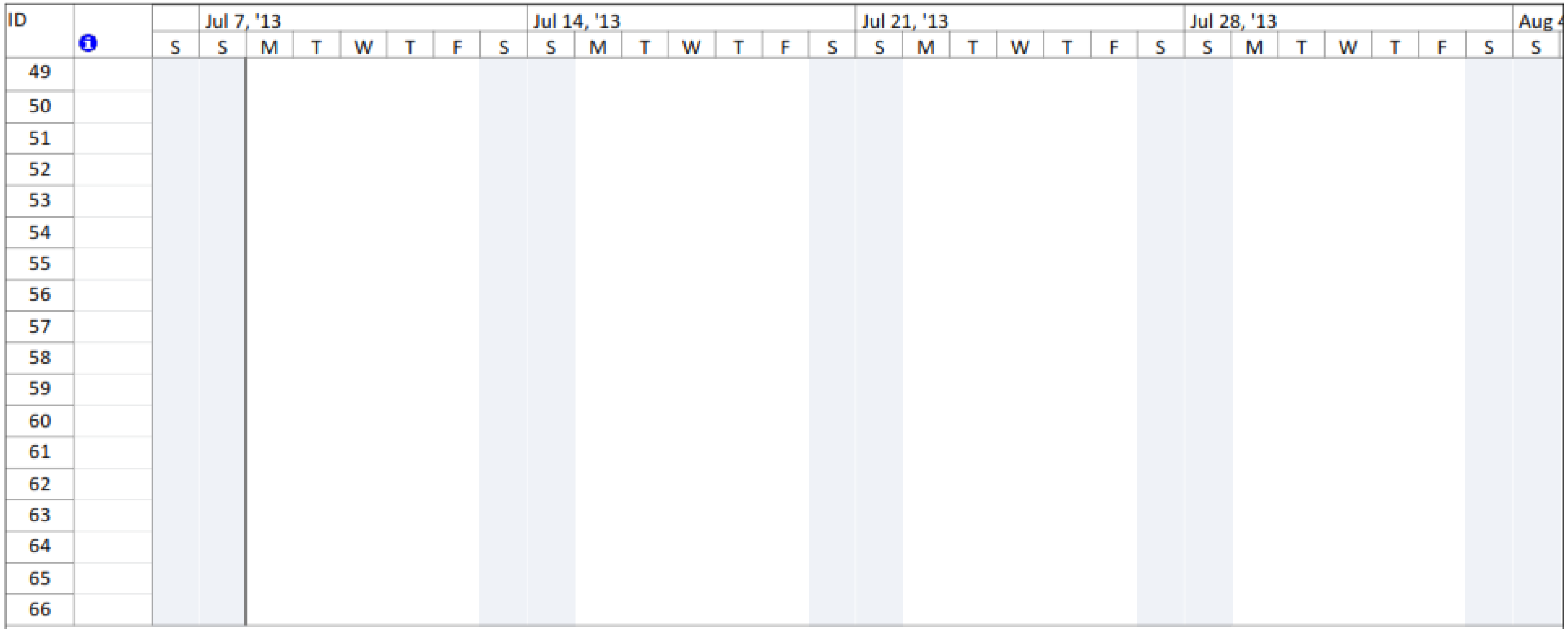
	ℹ	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names
44			**Testing Phase**	**17 days**	**Wed 9/11/13**	**Thu 10/3/13**	**2,13,29**	
45			Test Software	3 days	Wed 9/11/13	Fri 9/13/13		
46			Test Hardware	3 days	Wed 9/11/13	Fri 9/13/13		
47			Test Network	2 days	Wed 9/11/13	Thu 9/12/13		
48			Production Stress	14 days	Mon 9/16/13	Thu 10/3/13	45,46,47	
49			**Documentation Phase**	**2 days**	**Fri 10/4/13**	**Mon 10/7/13**	**44**	
50			**Document Software**	**2 days**	**Fri 10/4/13**	**Mon 10/7/13**		
51			Switch Configuration	1 day	Fri 10/4/13	Fri 10/4/13		
52			Router Configuration	1 day	Fri 10/4/13	Fri 10/4/13		
53			Server Configuration	2 days	Fri 10/4/13	Mon 10/7/13		
54			**Document Source Code**	**2 days**	**Fri 10/4/13**	**Mon 10/7/13**		
55			Temperature	1 day	Fri 10/4/13	Fri 10/4/13		
56			Humidity	1 day	Fri 10/4/13	Fri 10/4/13		
57			PiFace Software	1 day	Fri 10/4/13	Fri 10/4/13		
58			OpenNMS Plugins	2 days	Fri 10/4/13	Mon 10/7/13		
59			**Document Hardware**	**1 day**	**Fri 10/4/13**	**Fri 10/4/13**		
60			**Document Wiring**	**1 day**	**Fri 10/4/13**	**Fri 10/4/13**		
61			Cat 5 Wiring	1 day	Fri 10/4/13	Fri 10/4/13		
62			Sensor Schematics	1 day	Fri 10/4/13	Fri 10/4/13		
63			**Sensor Locations**	**1 day**	**Fri 10/4/13**	**Fri 10/4/13**		
64			Temperature	1 day	Fri 10/4/13	Fri 10/4/13		
65			Humidity	1 day	Fri 10/4/13	Fri 10/4/13		
66			Piface	1 day	Fri 10/4/13	Fri 10/4/13		

Time Line


ID
Jul 7, '13
Jul 14, '13
Jul 21, '13
Jul 28, '13
Aug 4
S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
Andy,Eligio,Omar
Andy,Eligio,Omar
Andy
Andy
Omar
Andy,Eligi
Andy,Eligi
Project: Raspitor Project RV2
Date: Sun 10/20/13
Task
Split
Milestone
Summary
Project Summary
External Tasks
External Milestone
Inactive Task
Inactive Milestone
Inactive Summary
Manual Task
Duration-only
Manual Summary Rollup
Manual Summary
Start-only
Finish-only
Deadline
Progress

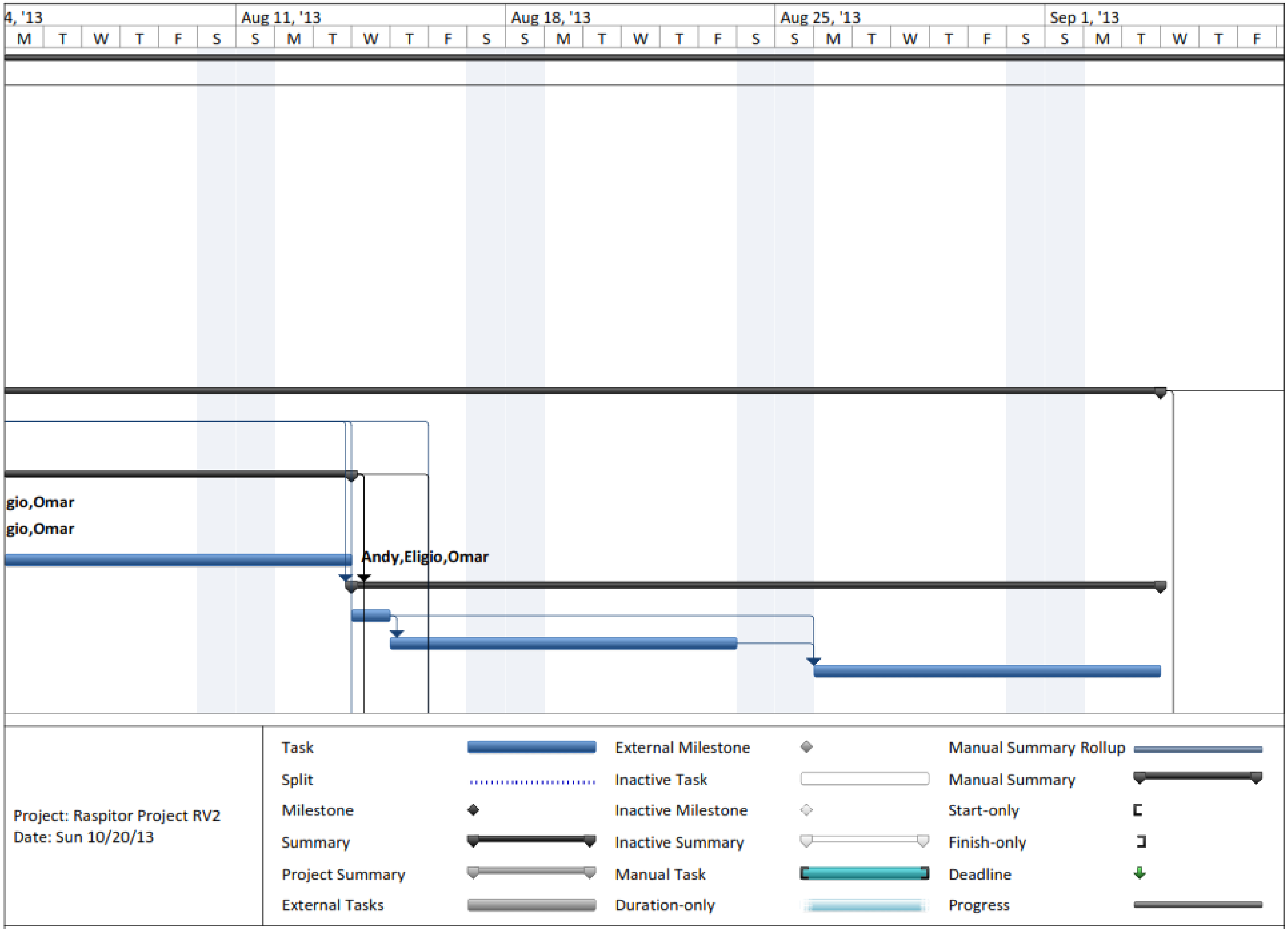


Project: Raspitor Project RV2
Date: Sun 10/20/13

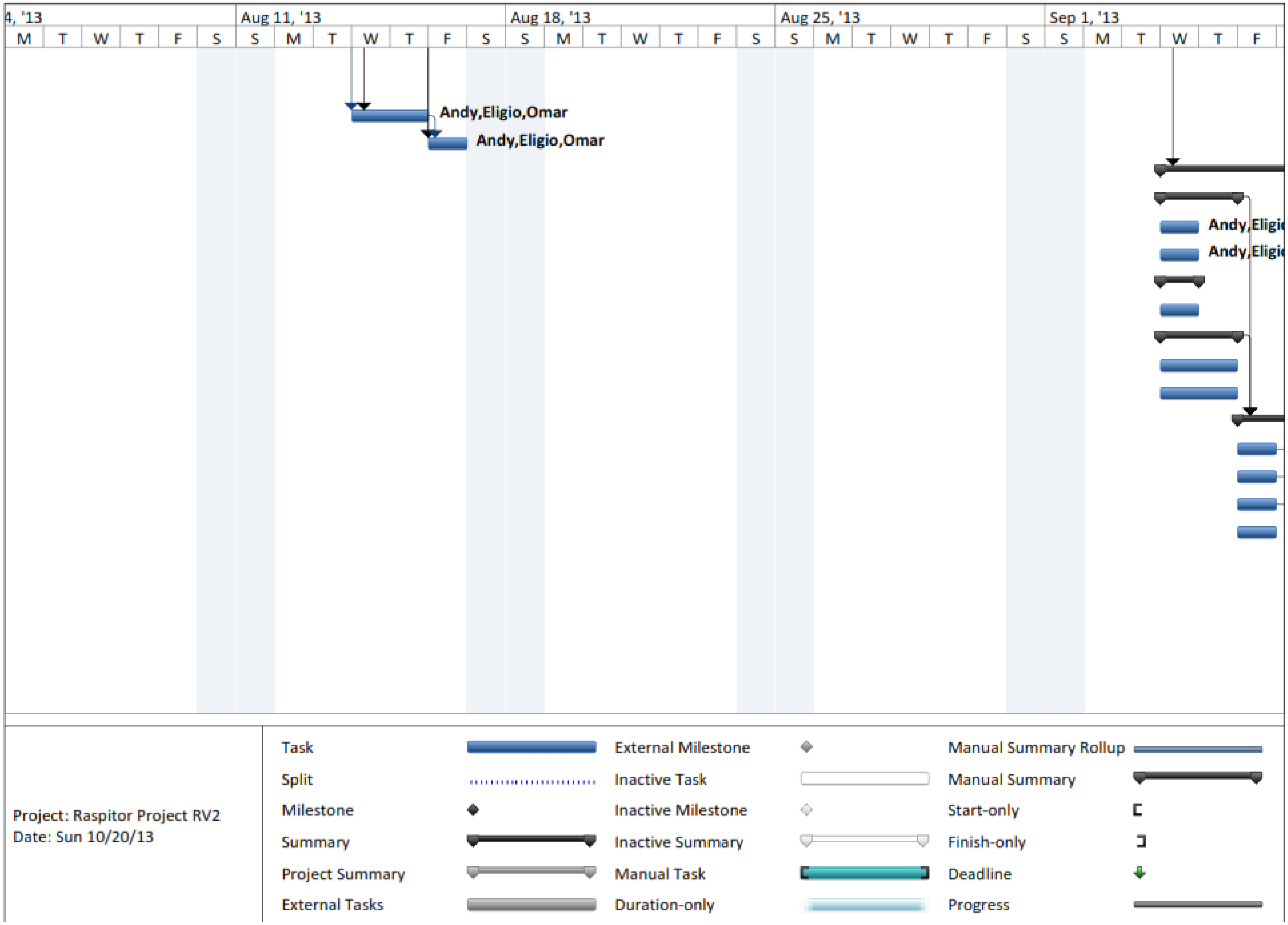


Project: Raspitor Project RV2
Date: Sun 10/20/13

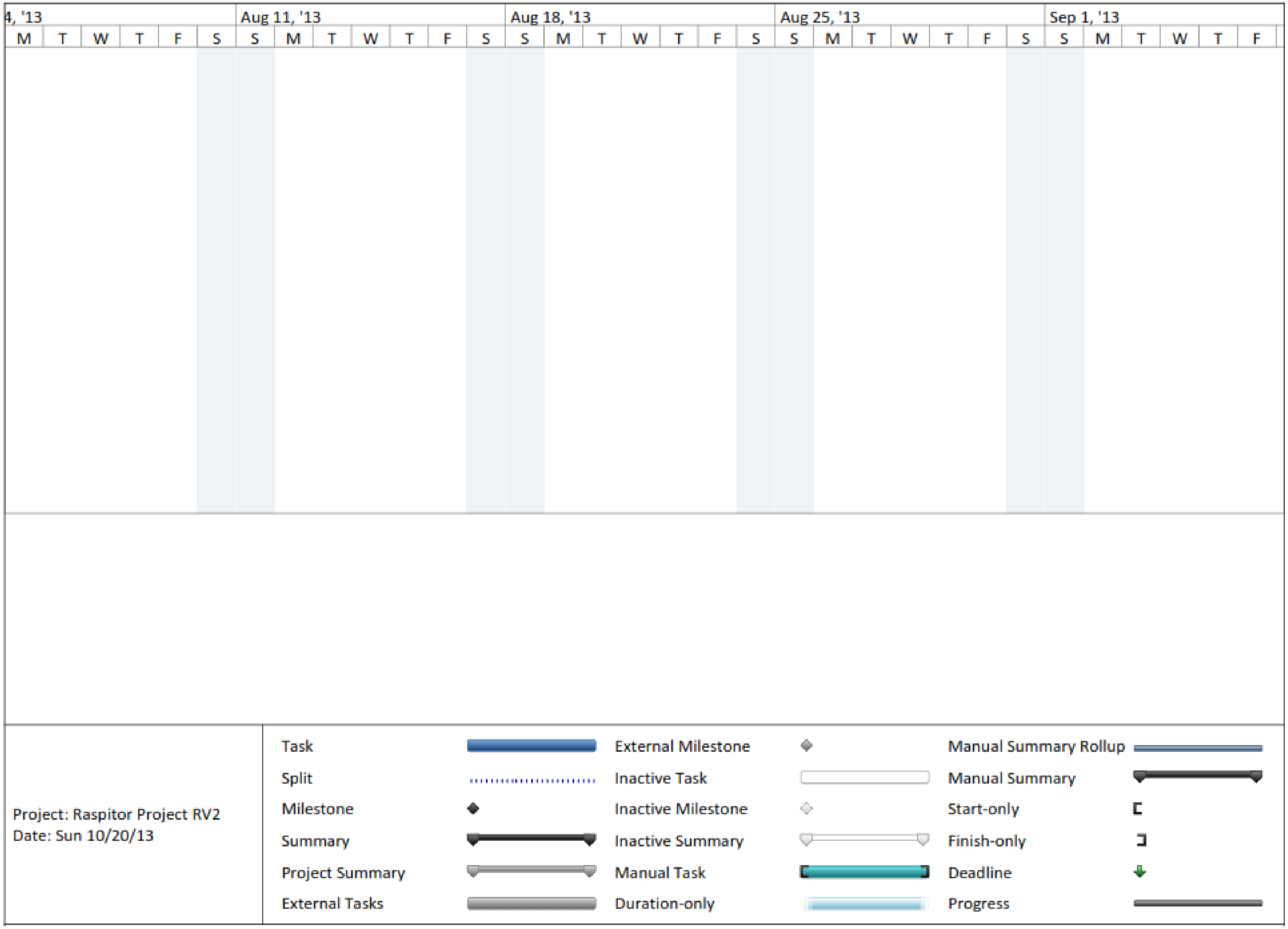
Task	▬	External Milestone	◆	Manual Summary Rollup	▬
Split	┈	Inactive Task	▭	Manual Summary	▬
Milestone	◆	Inactive Milestone	◇	Start-only	[
Summary	▬	Inactive Summary	▭	Finish-only	]
Project Summary	▬	Manual Task	▬	Deadline	⬇
External Tasks	▬	Duration-only	▬	Progress	▬



Project: Raspitor Project RV2
Date: Sun 10/20/13

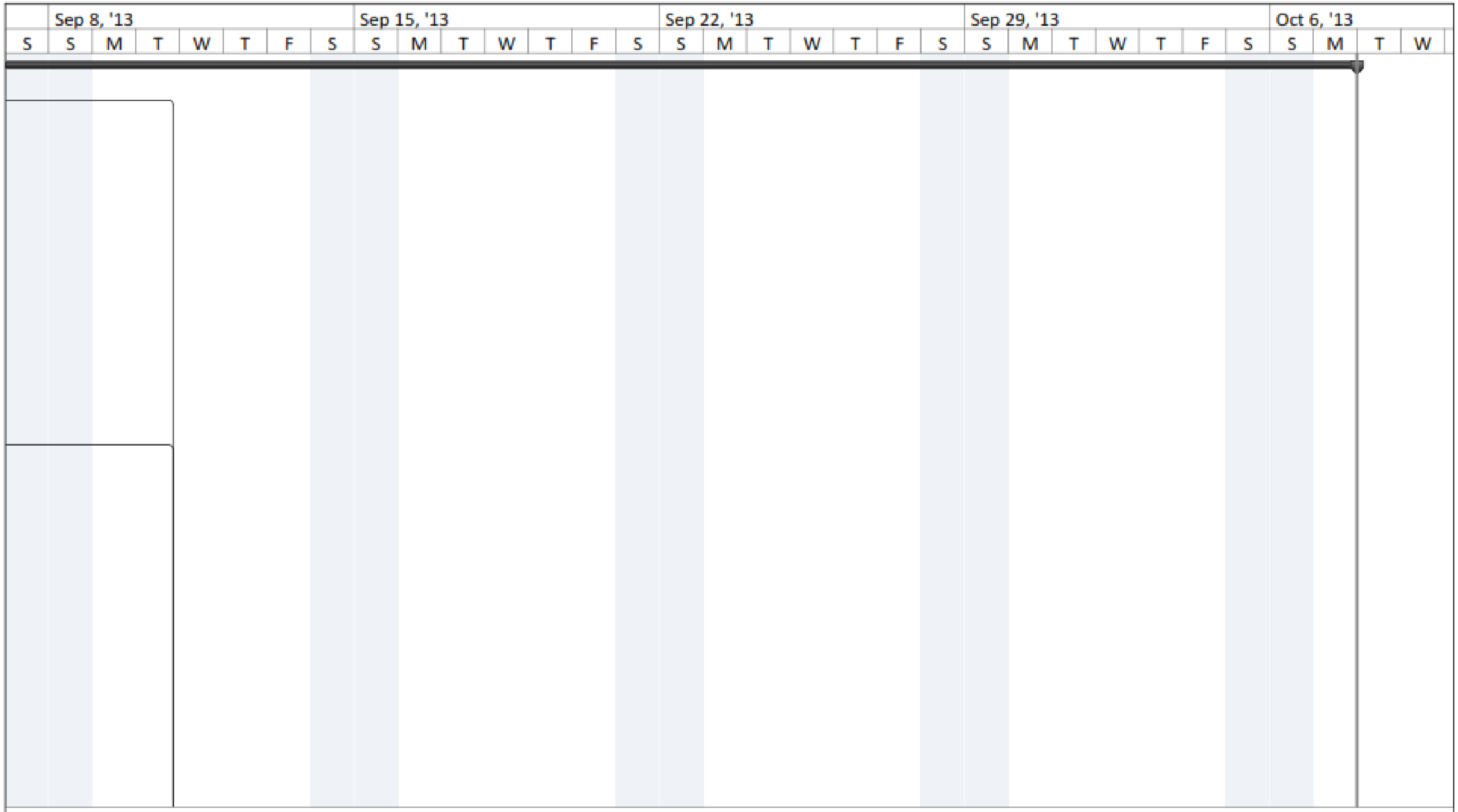


Project: Raspitor Project RV2
Date: Sun 10/20/13



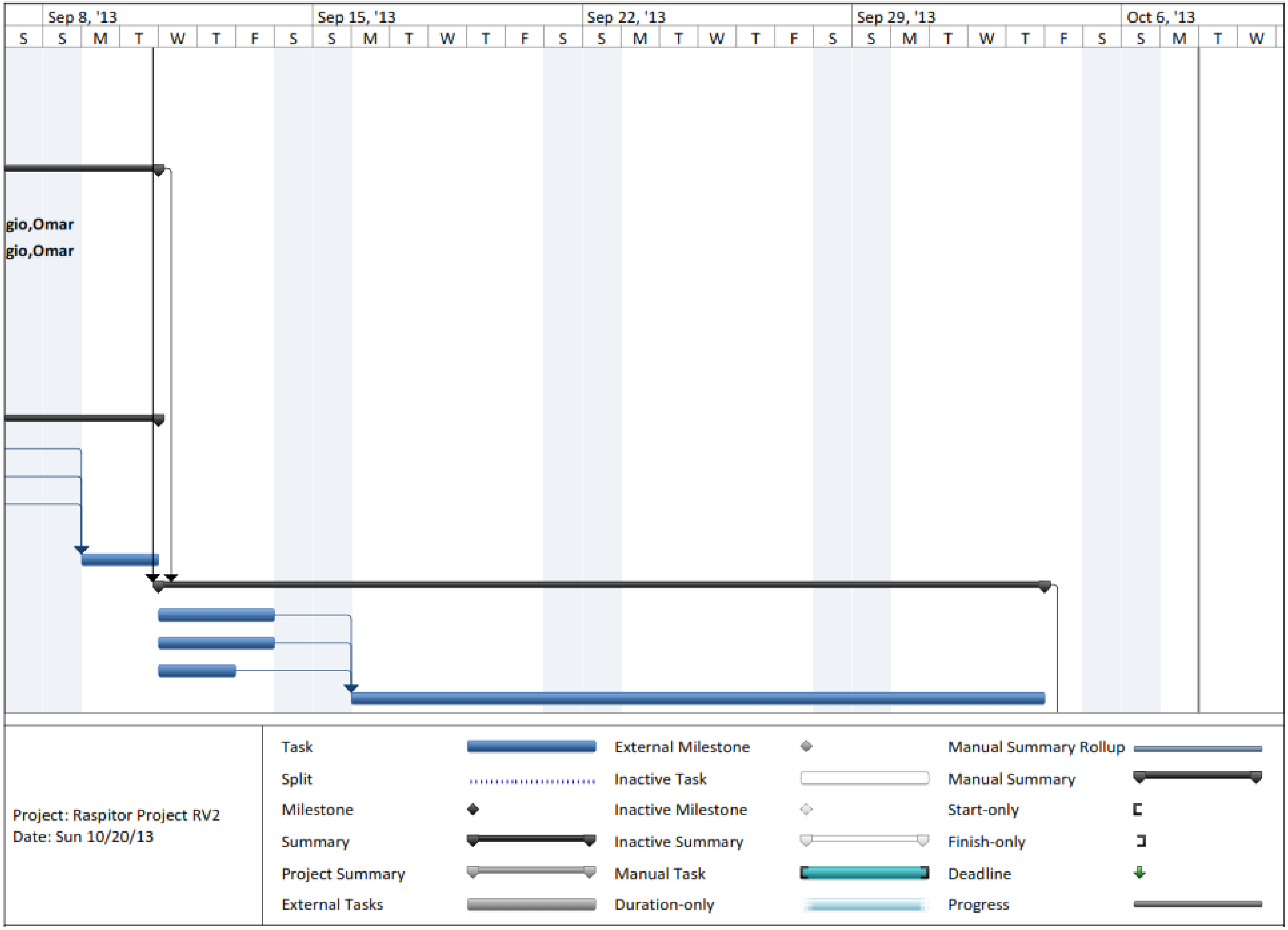
Project: Raspitor Project RV2
Date: Sun 10/20/13

Task	▬	External Milestone	◆	Manual Summary Rollup	▬
Split	⋯	Inactive Task	▭	Manual Summary	▬
Milestone	◆	Inactive Milestone	◇	Start-only	[
Summary	▬	Inactive Summary	▭	Finish-only	]
Project Summary	▬	Manual Task	▬	Deadline	⇩
External Tasks	▬	Duration-only	▬	Progress	▬

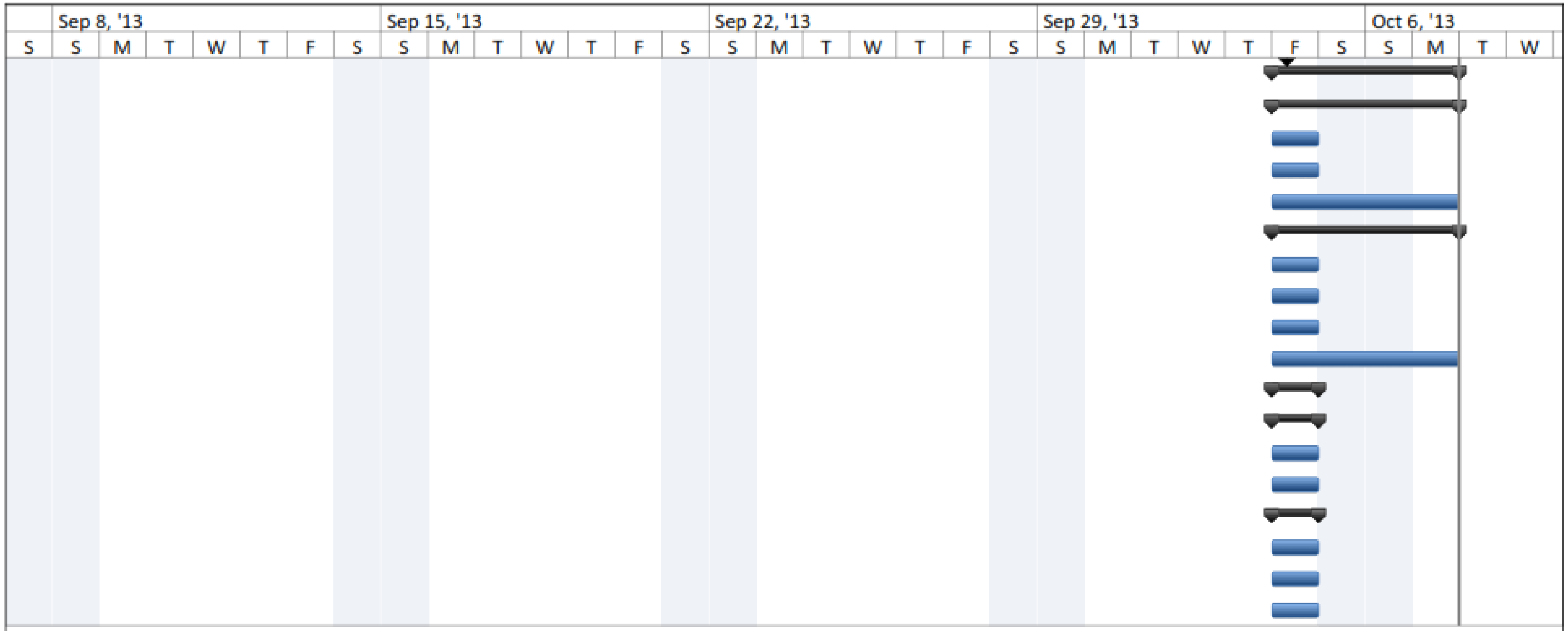


Project: Raspitor Project RV2
Date: Sun 10/20/13

Task	▬	External Milestone	◆	Manual Summary Rollup	▬
Split	┈	Inactive Task	▭	Manual Summary	▬
Milestone	◆	Inactive Milestone	◇	Start-only	[
Summary	▬	Inactive Summary	▭	Finish-only	]
Project Summary	▬	Manual Task	▬	Deadline	⇩
External Tasks	▬	Duration-only	▬	Progress	▬



Project: Raspitor Project RV2
Date: Sun 10/20/13



Project: Raspitor Project RV2
Date: Sun 10/20/13

Task	▬	External Milestone	◆	Manual Summary Rollup	▬
Split	┈	Inactive Task	▭	Manual Summary	▬
Milestone	◆	Inactive Milestone	◇	Start-only	[
Summary	▬	Inactive Summary	▭	Finish-only	]
Project Summary	▬	Manual Task	▬	Deadline	⇩
External Tasks	▬	Duration-only	▬	Progress	▬

References

AdaFruit Industries. (n.d.). *Raspberry Pi.* Retrieved on August 22, 2013 from http://www.adafruit.com/category/105

APC by Schneider Electric. (n.d.). NetBotz 200. Environmental monitoring for network closets to data centers. Retrieved on August 27, 2013 from http://www.apc.com/products/family/?id=346

Avatech. (2013). *Room Alert: IT & Facilities Environment Monitoring Product.* Retrieved from http://avtech.com/Products/Environment_Monitors/

Raspberry Pi Organization. (n.d.). *Raspberry Pi FAQs.* Retrieved on August 25, 2013 from http://www.raspberrypi.org/faqs

Reuk.co.uk.(2013). *Connect Multiple Temperature Sensors with Raspberry Pi*. Retrieved from http://www.reuk.co.uk/Connect-Multiple-Temperature-Sensors-with-Raspberry-Pi.htm

APC by Schneider Electric. (n.d.). NetBotz 200. Environmental monitoring for network closets to data centers. Retrieved on August 27, 2013 from http://www.apc.com/products/family/?id=346

Appendix

Product Specifications

Raspberry Pi

Raspberry Pi Probe:

Operating System: Raspbian

Memory: 512 MB ram

Processor: ARM11 @ 700 MHz

Video: Video core 4 GPU

Peripherals: Two DS18B20 temperature sensors DHT11 Humidity Sensor

Storage: 16GB SD card

Raspberry Pi Face:

Operating System: Raspbian

Memory: 512 MB ram

Processor: ARM11 @ 700 MHz

Video: Video core 4 GPU

Peripherals: Pi-Rack 3 Pi-Face Daughterboard

Storage: 16GB SD card

Raspberry LCD:

Operating System: Raspbian

Memory: 256 MB ram

Processor: ARM11 @ 700 MHz

Peripherals: HD44780 16 char x 2 lines LCD screen

Storage: 16GB SD

DS18B20 Temperature sensor:

Manufacturer: Maxim IC

Resolution: 9-12 Bit precision

Range: -55C to 125C (+/-0.5C)

Communication: 1 Wire Interface

DHT11 Humidity sensor:

Manufacturer: Unknown

Resolution: 8 Bit

Range: Temperature: 0-50C (+/- 2 C)

Humidity: RH 20-80 %(+/-5%RH)

Pi-Face Daughter board:

Manufacturer: PIFACE

Features:

- 2 relays
- 4 tactile switches
- 8 open-collector outputs
- 8 LED indicators

Servers:

DHCP Sever:

Dell PowerEdge 2650

Web Server:

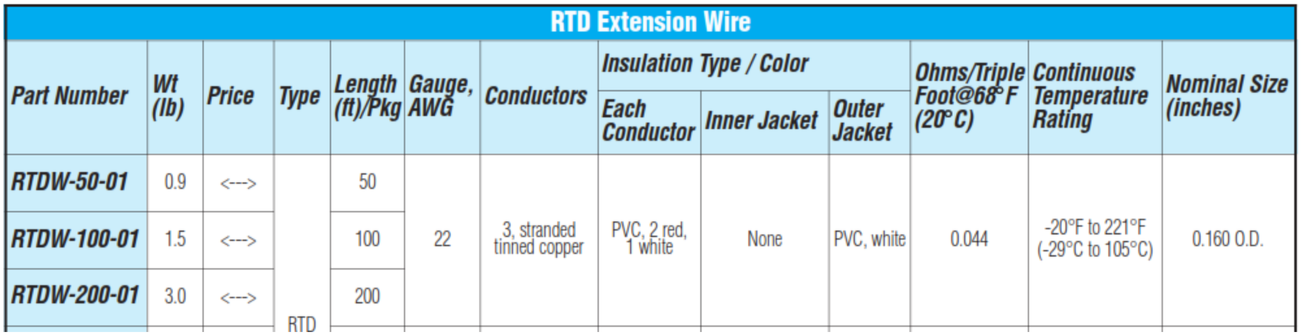
HP Proliant Server

Firewall Server:

Dell PowerEdge 2650

Cabling:

Wire 3-Conductor PVC Insulated 22GA Stranded



RTD Extension Wire

Part Number	Wt (lb)	Price	Type	Length (ft)/Pkg	Gauge, AWG	Conductors	Insulation Type / Color			Ohms/Triple Foot@68°F (20°C)	Continuous Temperature Rating	Nominal Size (inches)
							Each Conductor	Inner Jacket	Outer Jacket			
RTDW-50-01	0.9	<--->	RTD	50	22	3, stranded tinned copper	PVC, 2 red, 1 white	None	PVC, white	0.044	-20°F to 221°F (-29°C to 105°C)	0.160 O.D.
RTDW-100-01	1.5	<--->		100								
RTDW-200-01	3.0	<--->		200								

Additional Equipment

Triangle Engineering 24" Oscillating Vertical Mount Fan With Poly Housing 245546 1 HP 5900 CFM

Product Specifications

FAN DIAMETER INCHES	24
COLOR FINISH	Gray
ASSEMBLY	Unassembled
CONSTRUCTION	Polypropylene Head / Steel Mount
CFM HIGH	5,900
RPM	1,140
VOLTAGE	115 / 230
PHASE	1
AMPS	10.6/5.3
HORSEPOWER	1
FAN TYPE	Tiltable Fan Head
MOUNT TYPE	Vertical Bracket
TYPE	Oscillating
DESCRIPTION	24" High Velocity Fan
APPROVAL	OSHA Approved
MANUFACTURER'S PART NUMBER	245546
MODEL	245546

Ebac AD850E Dehumidifier

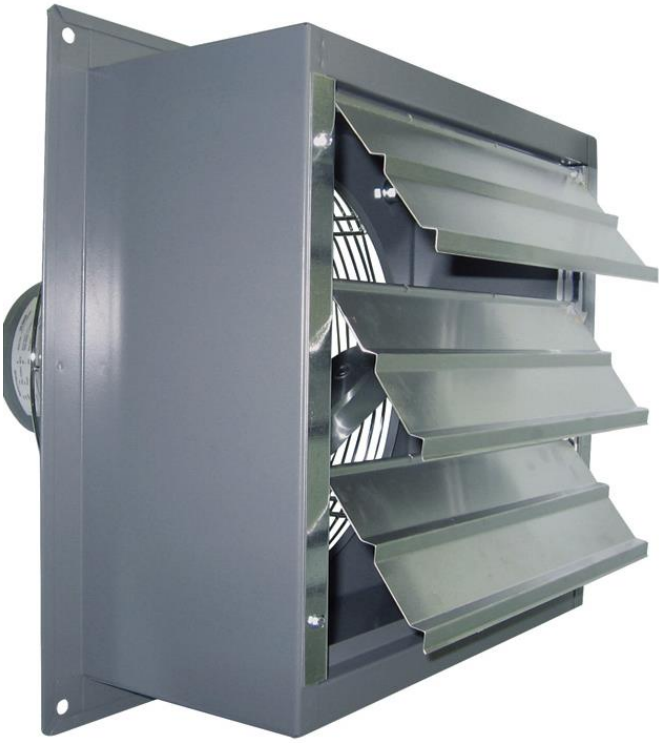
Technical Specifications of Ebac AD850 Dehumidifier

	Ebac AD850E Dehumidifiers
Coverage Area	Up to 1700 sq ft
Water Removal Capacity	87 pints per day set @ 80°F, 60%RH, 20 gallons at saturation
Operating Temperature Range	41°F - 95°F
CFM	300 cfm (max)
Noise Level	51 dba
Refrigerant Type	R407C (26 oz)
Power Supply	880 watts, 7.8 amps, 115V, 60Hz
Dimensions	27 1/8"H x 11"D x 26 3/4"W
Weight (lbs)	80
Warranty	1 yr. electrical. 3 yrs. compressor, condenser, evaporator

Canarm Wall Exhaust Fan

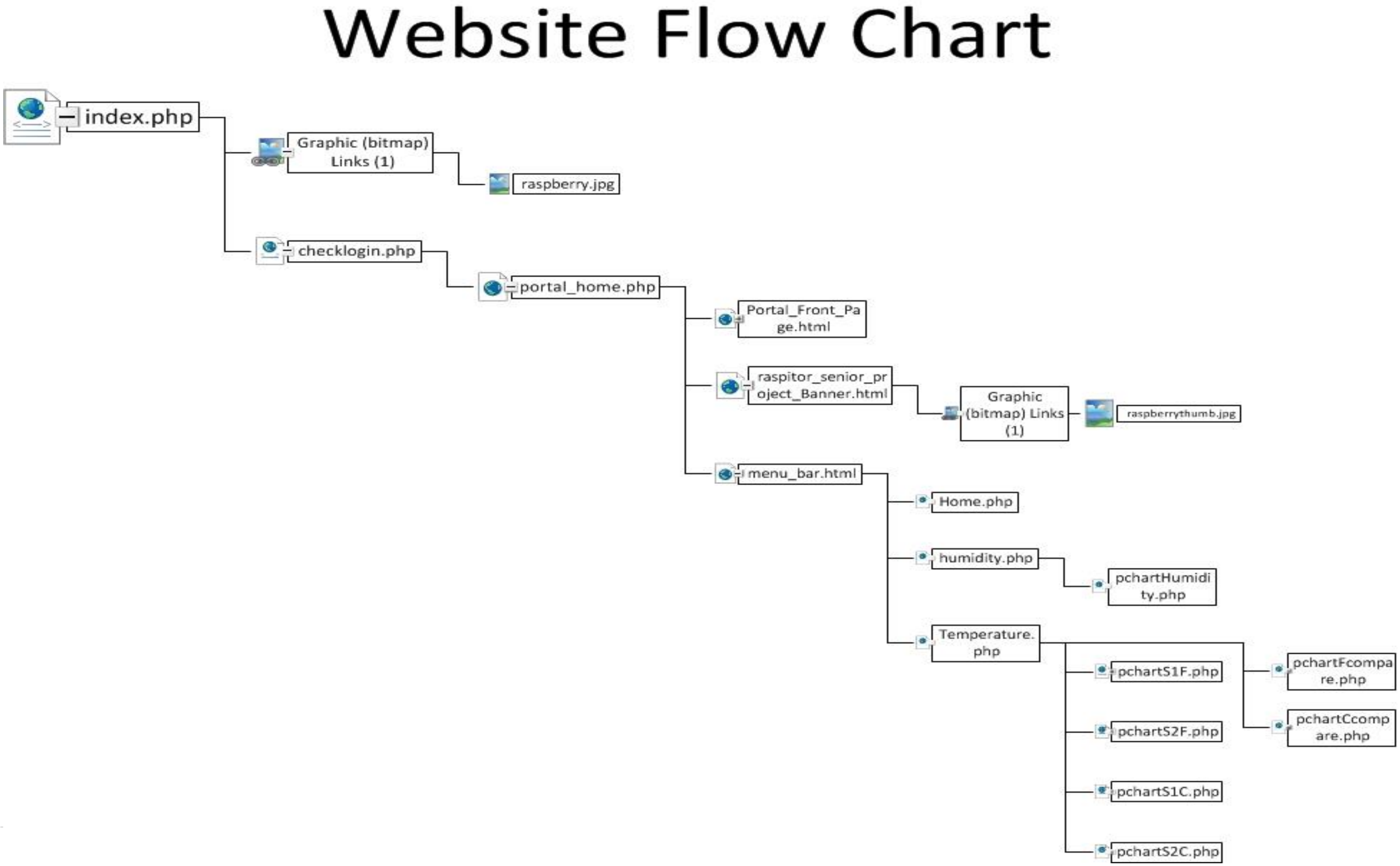
Key Specs

Item#	14757
Ship Weight	41.0 lbs
Fan Diameter (in.)	20
HP	1/3
Volts	115
Needs to be Hardwired	Yes

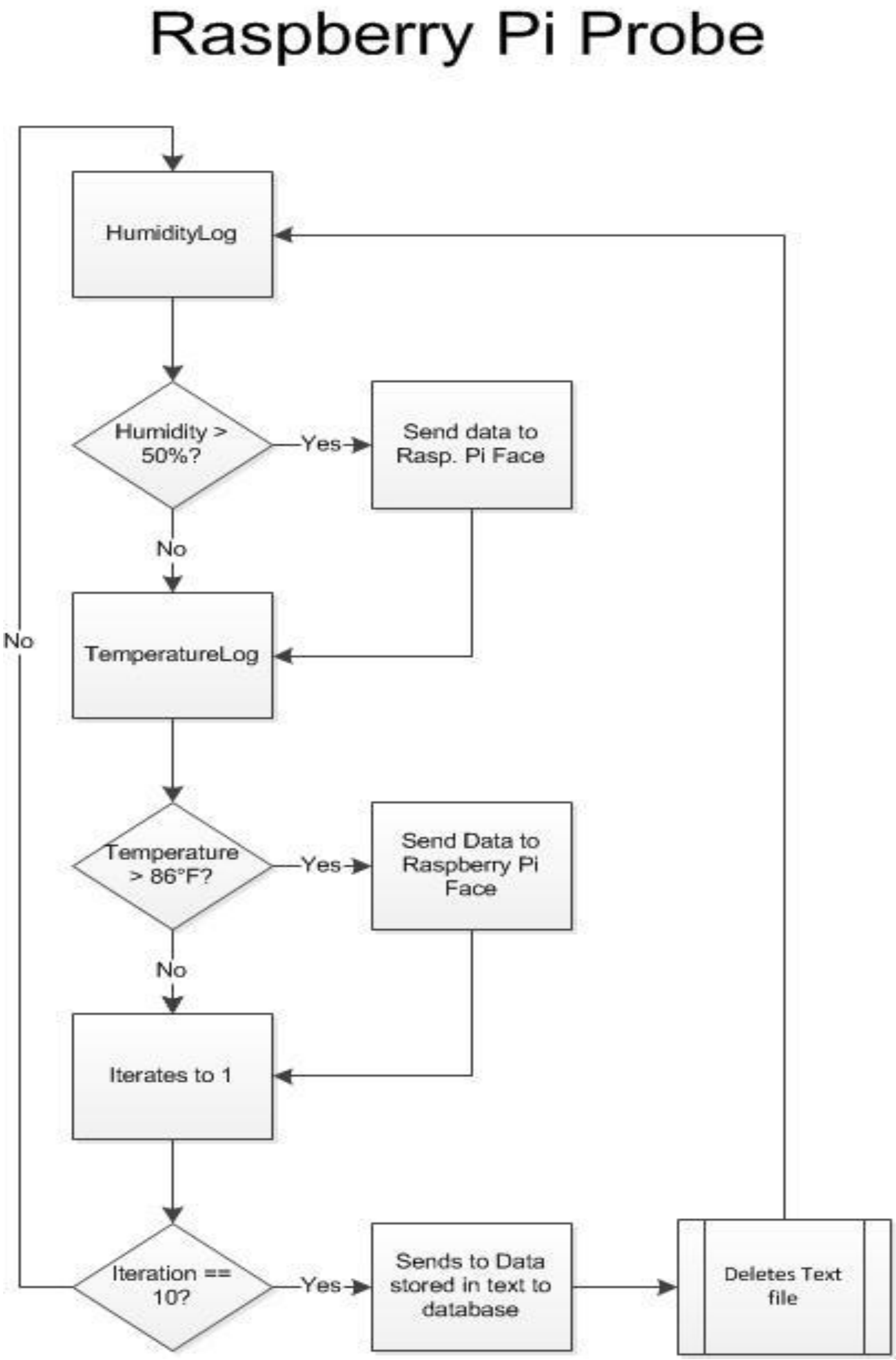


SOFTWARE FLOW CHARTS

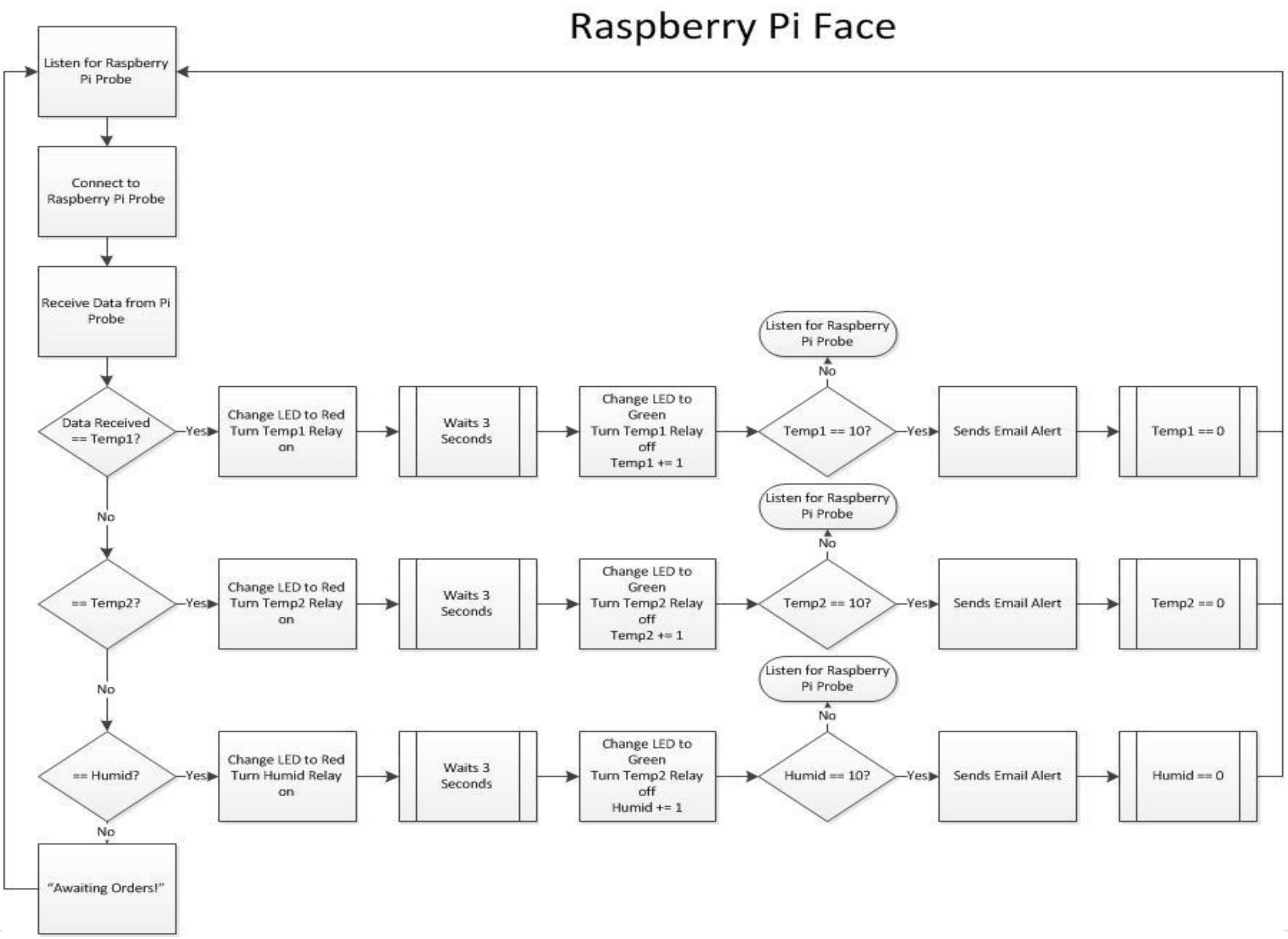
Website Flow Chart



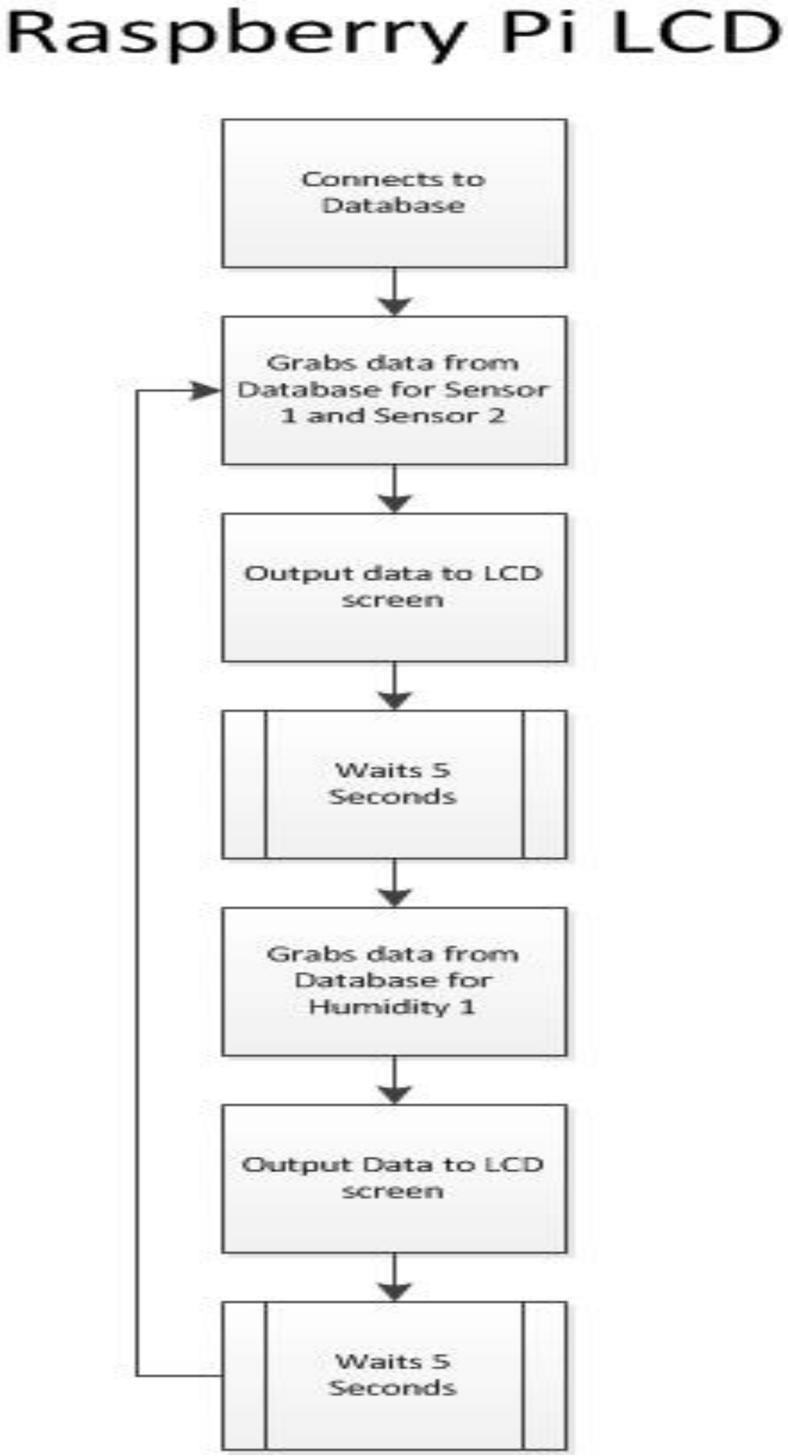
Raspberry PI Probe Flow Chart



Raspberry Pi Face Flow Chart


Raspberry Pi Face
Listen for Raspberry Pi Probe
Connect to Raspberry Pi Probe
Receive Data from Pi Probe
Data Received == Temp1?
Yes
Change LED to Red Turn Temp1 Relay on
Waits 3 Seconds
Change LED to Green Turn Temp1 Relay off Temp1 += 1
Temp1 == 10?
No
Listen for Raspberry Pi Probe
Yes
Sends Email Alert
Temp1 == 0
No
== Temp2?
Yes
Change LED to Red Turn Temp2 Relay on
Waits 3 Seconds
Change LED to Green Turn Temp2 Relay off Temp2 += 1
Temp2 == 10?
No
Listen for Raspberry Pi Probe
Yes
Sends Email Alert
Temp2 == 0
No
== Humid?
Yes
Change LED to Red Turn Humid Relay on
Waits 3 Seconds
Change LED to Green Turn Temp2 Relay off Humid += 1
Humid == 10?
No
Listen for Raspberry Pi Probe
Yes
Sends Email Alert
Humid == 0
No
"Awaiting Orders!"

Raspberry Pi LCD Flow Chart


Raspberry Pi LCD
Connects to Database
Grabs data from Database for Sensor 1 and Sensor 2
Output data to LCD screen
Waits 5 Seconds
Grabs data from Database for Humidity 1
Output Data to LCD screen
Waits 5 Seconds

PROGRAMMING CODES

Raspberry Pi Probe:

HumidityLogging.py:

```
import subprocess
import re
import time
import datetime

Iteration = 0

def time_stamp():
    ts = time.time()
    st = datetime.datetime.fromtimestamp(ts).strftime('%Y-%m-%d %H:%M:%S')
    return st

for humiditylogging in range (10000):
#   while True:
    output = subprocess.check_output(["./Adafruit_DHT", "11", "23"]);
    #print output
    #print time_stamp()

    #Searches the for the value of Temperature
    matches = re.search("Temp =\s+([0-9.]+)", output)
    if (not matches):
        time.sleep(3)
        continue
    temperature = float(matches.group(1))
#   print temperature

    #Searches for the value of Humidity
    matches = re.search("Hum =\s+([0-9.]+)", output)
    if (not matches):
        time.sleep(3)
        continue
    humidity = float(matches.group(1))
#   print humidity

    temp = str(temperature)
    humid = str(humidity)

#   print "Temperature: .1f C" + temp
#   print "Humidity: %.1f %%" + humid
    with open('/ram/humiditylogging.txt', 'a') as log:
        humidlog = str(temp) + '\t' + str(humidity)
        log.write('NULL\t' + 'H1\t' + time_stamp() + '\t' + humidlog + "\n")
        print 'NULL\t' + 'H1\t' + time_stamp() + '\t' + humidlog
    log.close()
    execfile('TemperaturePi_Optimized_experimental_Quick_stable.py')
    time.sleep(3)

    Iteration += 1
    if Iteration == 10:
        print "Sorting data!"
        os.system('sudo python TemperatureSort.py')
        Iteration = 0
```

TemperaturePi_Optimized_experimental_Quick_stable.py:

```
import os
import glob
import time
import datetime
import socket

#Connection Info for the RaspberryPi Piface
#s = socket.socket()
#host = '192.168.1.246'
#port = 1133
#s.connect((host, port))

def time_stamp():
    ts = time.time()
    st = datetime.datetime.fromtimestamp(ts).strftime('%Y-%m-%d,%H:%M:%S')
    return st
# initiation of each devices
os.system('modprobe w1-gpio')
os.system('modprobe w1-therm')

def initialize():
    for sensors in range(0,2):
    #Base Directory
        base_dir = '/sys/bus/w1/devices/'
    #Grabs data from each of the sensors
        device_folder = glob.glob(base_dir + '28*')[sensors]
        device_file = device_folder + '/w1_slave' #file where the sensors are located
    #print device_file
    #Data into Text File
    #log = open("temperaturelog.txt", "a")
    #datalog = log.read()

    #grabs the data raw
    def read_temp_raw():
    f = open(device_file, 'r')
    lines = f.readlines()
        #lines2 = f.readlines()
    f.close()
    return lines
    #print read_temp_raw()
    #interpret each data

        def read_temp():
            output = read_temp_raw()
            while output[0].strip()[-3:] != 'YES':
                time.sleep(0.2)
                output = read_temp_raw()
            equals_pos = output[1].find('t=')
            if equals_pos != -1:
                temp_string = output[1][equals_pos+2:]
                temp_c = float(temp_string) / 1000.0
            return temp_c
    #Gathes the data from the if statement and stores it into each variable
        if sensors == 0:
            temp0 = read_temp()
            temp1 = read_temp()
        else:
            temp1 = read_temp()
    #print temp0, temp1
    return temp0, temp1
#Debug use Output
#print "If sensor"
#print temp0
#print temp1

for datalog in range (1):
    while True:
    #starts the tead_temp() function
        gatherdata = initialize()
        #print gatherdata #debug use

    #Stores it into data
    #s1c = Sensor 1 Celcius
    #s2c = Sensor 2 Celcius
        s1c, s2c = gatherdata
        #debug use
        #print s1c
        #print s2c
    #Conversion to F
```

```
        s1f = s1c * 1.8 + 32
        s2f = s2c * 1.8 + 32
    #Rounding
        s1fr = round(s1f, 3)
        s2fr = round(s2f, 3)
        #Debug Conversion
        #print s1f
        #print s2f
    #Convert the data into strings
        sen1c = str(s1c)
        sen1f = str(s1fr)
        sen2c = str(s2c)
        sen2f = str(s2fr)
        #Debug strings
        #print sen1c
        #print sen1f
        #print sen2c
        #print sen2f

        #print ('S1\t' + time_stamp() + '\t' + sen1c + '\t' + sen1f )
        #print ('S2\t' + time_stamp() + '\t' + sen2c + '\t' + sen2f )

        datalog = str('NULL\t' + 'S1\t' + time_stamp() + '\t' + sen1c + '\t' + sen1f)
        datalog1 = str('NULL\t' + 'S2\t' + time_stamp() + '\t' + sen2c + '\t' + sen2f)
        if s1f >= 86:
            print "HOT!, Need to turn the fan on!"
            s.send('Temp1')
            s.close()
        print datalog
        print datalog1
        with open('/ram/temperaturelog.txt', 'a') as log:
            log.write(datalog+'\n')
            log.write(datalog1+'\n')
        log.close()
        break
```

TemperatureSort.py:

```
import csv
import os

temp_file = r"/ram/temperaturelog.txt"
csv_temp_file = r"/ram/temperaturelog.csv"
text_file = r"/ram/humiditylogging.txt"
csv_file = r"/ram/humiditylogging.csv"
```

```
def sort():
	in_text = csv.reader(open(text_file, "rb"), delimiter = '\t')
	out_csv = csv.writer(open(csv_file, "wb"))
	out_csv.writerows(in_text)
	temp_text = csv.reader(open(temp_file, "rb"), delimiter = '\t')
	out_temp_csv = csv.writer(open(csv_temp_file, "wb"))
	out_temp_csv.writerows(temp_text)
def execute():
	os.system('sudo python Database.py')
	os.system('sudo rm /ram/temperaturelog.txt')
	os.system('sudo rm /ram/temperaturelog.csv')
	os.system('sudo rm /ram/humiditylogging.txt')
	os.system('sudo rm /ram/humiditylogging.csv')

sort()
execute()
#os.system('sudo rm /ram/temperaturelog.txt')
#os.system('sudo rm /ram/temperaturelog.csv')
#os.system('sudo rm /ram/humiditylogging.txt')
#os.system('sudo rm /ram/humiditylogging.csv')
```

Databse.py:

```
#!/usr/bin/python

import MySQLdb			#imports the MYSQL database
import csv			#imports the csv function
import time
start_time = time.time()

db = MySQLdb.connect("192.168.1.240", "raspberrysensor", "raspberry", "raspitor", local_infile
= 1)

cursor = db.cursor()

temperature = """LOAD DATA LOCAL INFILE '/ram/temperaturelog.csv' INTO TABLE
temperature FIELDS TERMINATED BY ',' ENCLOSED BY '"' LINES TERMINATED BY
'\n';"""

humidity = """LOAD DATA LOCAL INFILE '/ram/humiditylogging.csv' INTO TABLE
humidity FIELDS TERMINATED BY ',' ENCLOSED BY '"' LINES TERMINATED BY '\n';"""

try:
	#executes the SQL commands
	cursor.execute(temperature)
	cursor.execute(humidity)
	print "Humid in"
```

```
    #Changes are Comitted into database
    db.commit()
    print "win"
except:
    #Rolls Back if there is error
    db.rollback()
    print "Lose"
db.close()
stop_time = time.time()
execute = start_time - stop_time
print execute
```

Raspberry Pi-Face:

Raspitor.py:

```
import pifacedigitalio as pfio
import socket
from time import sleep
pfio.init()
pfio.init(False)
s = socket.socket()
host = "192.168.1.246"#socket.gethostname()
port = 1133
print host
TemperatureRelay1 = 0
TemperatureRelay2 = 0
HumidityRelay1 = 0

#sensor #1
def temperature1():
    print "Temperature 1 Relay ON!"
    pfio.digital_write(0,1) #Turns on relay #1
    pfio.digital_write(2,0) #Turns off LED Green
    pfio.digital_write(3,1) #Turns on LED Red
    sleep(3)
    print "Temperature 1 Relay Off :("
    pfio.digital_write(0,0) #Turn off relay #1
    pfio.digital_write(2,1) #Turn on LED green
    pfio.digital_write(3,0) #Turns red LED red

#sensor #2
def temperature2():
    print "Temperature 2 Relay ON!"
    pfio.digital_write(1,1)#Turns on Relay #2
    pfio.digital_write(4,0) #Turns off LED Green
    pfio.digital_write(5,1) #Turns on LED red
    sleep(3)
    print "Tempearture 2 Relay off :("
    pfio.digital_write(1,0)#Turns off Relay $2
    pfio.digital_write(4,1)#Turns on LED Green
    pfio.digital_write(5,0) #Turns off LED Red

#Humidity Sensor
def humidity():
    print "Humidity 2 Relay ON!"
    pfio.digital_write(0,1,1) #Turns on Relay #1 Board #1
    pfio.digital_write(2,0,1) #Turns off LED green Board #1
    pfio.digital_write(3,1,1) #Turns on LED red Board #1
    sleep(3)
    print "Humidity 2 Relay Off :("
    pfio.digital_write(0,0,1) #Turns off Relay #1 Board #1
    pfio.digital_write(2,1,1) #Turns on LED green Board #1
    pfio.digital_write(3,0,1) #Turns off LED red board #1
s.bind((host, port))
s.listen(5)
while True:
    c, addr = s.accept()
    print 'Got connection from', addr
    device = c.recv(1024)
    if device == 'Temp1':
        temperature1()
        TemperatureRelay1 += 1
    elif device == 'Temp2':
        temperature2()
        TemperatureRelay2 += 1
    elif device == 'Humid':
        humidity()
        HumidRelay1 += 1
    else:
        print "Awaiting orders!"

    if TemperatureRelay1 == 10 or TemperatureRelay2 == 10 or HumidityRelay1 == 10:
        print("ALERT! SENDING EMAIL")
        execfile("send.py")
        TemperatureRelay1 = 0
        TemperatureRelay2 = 0
        HumidityRelay1 = 0

    c.close()
```

Send.py:

```
    # coding: latin1
```

```
import time
import smtplib
import MySQLdb

TO = 'alau2@my.devry.edu'
GMAIL_USER = 'RaspitorPi@gmail.com'
GMAIL_PASS = 'raspberryseniorproject'

SUBJECT = 'ALERT!!'

def grab_temp_data():
        #Login Credentials
db = MySQLdb.connect("192.168.1.240", "raspberrysensor", "raspberry", "raspitor")
cursor = db.cursor()

        #self Explanitory
        #SQL Statement is Select the average temperature from the last 10 recent entry made
by date.
        GrabTemperatureS1 = """SELECT avg( `fahrenheit` )
            FROM (SELECT * FROM `temperature`
            WHERE `sensors` = 'S1'
            ORDER BY `temperature`.`date` DESC LIMIT 0 , 10)s"""

        GrabTemperatureS2 = """SELECT avg( `fahrenheit` )
        FROM (SELECT * FROM `temperature`
        WHERE `sensors` = 'S2'
        ORDER BY `temperature`.`date` DESC LIMIT 0 , 10)s"""

        #Converts the statement made by the SQL into numbers to display on LCD
cursor.execute(GrabTemperatureS1)
data = cursor.fetchall()
for row in data:
                TempS1 = round(row[0],3)
        #debug use
        #print TempS1

cursor.execute(GrabTemperatureS2)
data = cursor.fetchall()
for row in data:
                TempS2 = round(row[0],3)
        #debug use
        #print TempS2

        #Close Database
db.close()
            TS1 = str(TempS1)
```

```
      TS2 = str(TempS2)
return TS1, TS2

def grab_humid_data():
    #Login Credentials
db = MySQLdb.connect("192.168.1.240", "raspberrysensor", "raspberry", "raspitor")
cursor = db.cursor()
    #self Explanitory SQL Statement is Select the average temperature from the last 10
recent entry made by date.
    GrabHumidH1 = """SELECT avg(`humidity`)
    FROM (SELECT * FROM `humidity`
    WHERE `sensors` = 'H1'
    ORDER BY `humidity`.`date` DESC LIMIT 0,10)s"""

    #Converts the statement made by the SQL into numbers to display on LCD
cursor.execute(GrabHumidH1)
data = cursor.fetchall()
for row in data:
        HumidH1 = round(row[0],3)
    #debug use print TempS2 Close Database
db.close()
      Humidstring = str(HumidH1)
return Humidstring

email_data = grab_temp_data()
email_humid = grab_humid_data()
Temperature1, Temperature2 = email_data
Humid = email_humid
TEXT = "ALERT!! Your configured threshold exceeded!! \n Current Temps are: \n
Temperature Probe 1: " + Temperature1 + " °F \n Temperature Probe 2: " + Temperature2
+ " °F\n"
TEXT2 = "Current Humidity are: " + Humid + " %\n\n"
def send_email():
      print("Sending Email")
      smtpserver = smtplib.SMTP("smtp.gmail.com",587)
      smtpserver.ehlo()
      smtpserver.starttls()
      smtpserver.ehlo
      smtpserver.login(GMAIL_USER, GMAIL_PASS)
      header = 'To:' + TO + '\n' + 'From: ' + GMAIL_USER
      header = header + '\n' + 'Subject:' + SUBJECT + '\n'
      print header
      msg = header + '\n' + TEXT + TEXT2 + ' \n\n'
      smtpserver.sendmail(GMAIL_USER, TO, msg)
      smtpserver.close()
```

```
send_email()
```

Raspberry LCD:

RaspitorDisplay_updates.py:

```
#!/usr/bin/python

from Adafruit_CharLCD import Adafruit_CharLCD
from subprocess import *
from time import sleep, strftime
from datetime import datetime
import MySQLdb
import RPi.GPIO as GPIO
import time
lcd = Adafruit_CharLCD()

#This functions grabs the data from the MYSQL DATABASE
def grabtempdata():
    #Login Credentials
    db = MySQLdb.connect("192.168.1.240", "raspberrysensor", "raspberry", 
"raspitor")
    cursor = db.cursor()

    #self Explanitory
    #SQL Statement is Select the average temperature from the last 10 recent entry 
made by date.
    GrabTemperatureS1 = """ SELECT avg(`farenheight`)
    FROM (SELECT * FROM `temperature`
    WHERE `sensors` = 'S1'
    ORDER BY `temperature`.`date` DESC LIMIT 0,10)s"""

    GrabTemperatureS2 = """ SELECT avg(`farenheight`)
    FROM (SELECT * FROM `temperature`
    WHERE `sensors` = 'S2'
    ORDER BY `temperature`.`date` DESC LIMIT 0,10)s"""

    #Converts the statement made by the SQL into numbers to display on LCD
    cursor.execute(GrabTemperatureS1)
    data = cursor.fetchall()
    for row in data:
        TempS1 = round(row[0],3)
    #debug use
    #print TempS1

    cursor.execute(GrabTemperatureS2)
    data = cursor.fetchall()
    for row in data:
        TempS2 = round(row[0],3)
    #debug use
    #print TempS2

    #Close Database
    db.close()
    return TempS1, TempS2

def grabhumidata():
    #Login Credentials
    db = MySQLdb.connect("192.168.1.240", "raspberrysensor", "raspberry",
"raspitor")
    cursor = db.cursor()
    #self Explanitory SQL Statement is Select the average temperature from the last 10
recent entry made by date.
    GrabHumidH1 = """SELECT avg(`humidity`)
   FROM (SELECT * FROM `humidity`
   WHERE `sensors` = 'H1'
   ORDER BY `humidity`.`date` DESC LIMIT 0,10)s"""

    #Converts the statement made by the SQL into numbers to display on LCD
    cursor.execute(GrabHumidH1)
    data = cursor.fetchall()
    for row in data:
        HumidH1 = round(row[0],3)
    #debug use print TempS2 Close Database
    db.close()
    return HumidH1

#LCD stuff
lcd.begin(16,1)
lcd.clear()
lcd.message('The Raspitor\n')
lcd.message('Project\n')
sleep(5)
lcd.clear()
lcd.message('Please Wait...\n')
lcd.message('Loading Database')
sleep(5)

while True:
    displaytempdata = grabtempdata()
displayhumiddata = grabhumidata()
   TS1, TS2 = displaytempdata
   HH1 = displayhumiddata
```

```
lcd.clear()
lcd.message('S1: %s F\n' % TS1)
        lcd.message('S2: %s F\n' % TS2)
        sleep(5)
        lcd.clear()
lcd.message('H1: %s' % HH1)
sleep(5)
```

Web Server:

Index.php:

```
<html>
<head>
<title>Raspitor System Login</title>
<style type="text/css">
.auto-style1 {
        margin-left: 200px;
        text-align: center;
}
.auto-style2 {
        vertical-align: middle;
}
.auto-style3 {
        font-family: Arial, Helvetica, sans-serif;
        font-size: x-large;
}
.auto-style4 {
        margin-left: 360px;
}
</style>
</head>

<body style="height: 355px">

<div style="width: 600px" class="auto-style1">
        <img alt="" class="auto-style2" height="150" src="images/raspberry.jpg"
width="150" /><br />
        <span class="auto-style3">Raspitor System Login</span></div>

<div style="width: 300px" class="auto-style4">
<table width="300" border="0" align="center" cellpadding="0" cellspacing="1"
bgcolor="#CCCCCC">
<tr>
<form name="form1" method="post" action="checklogin.php">
<td>
```

```
<table width="100%" border="0" cellpadding="3" cellspacing="1" bgcolor="#FFFFFF">
<tr>
<td colspan="3"><strong>Administrator Login </strong></td>
</tr>
<tr>
<td width="78">Username</td>
<td width="6">:</td>
<td width="294"><input name="myusername" type="text" id="myusername"></td>
</tr>
<tr>
<td>Password</td>
<td>:</td>
<td><input name="mypassword" type="text" id="mypassword"></td>
</tr>
<tr>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td><input type="submit" name="Submit" value="Login"></td>
</tr>
</table>
</td>
</form>
</tr>
</table>
</div>

</body>
```

```
</html>
```

Checklogin.php:

```
<?php

// Database information
$host="localhost";
$username="root";
$password="";
$db_name="raspitor";
$tbl_name="users";

// Connect to Server
mysql_connect("$host","$username","$password") or die("Cannot connect");
mysql_select_db("$db_name")or die("cannot select DataBase");

//Username and password sent from form
$myusername=$_POST['myusername'];
```

```
        $mypassword=$_POST['mypassword'];

        //To protect MySQL injection
        $myusername = stripslashes($myusername);
        $mypassword = stripslashes($mypassword);
        $myusername = mysql_real_escape_string($myusername);
        $mypassword = mysql_real_escape_string($mypassword);
        $sql = "select * from $tbl_name users WHERE username='$myusername' and
        password='$mypassword'";
        $result = mysql_query($sql);

        // Counting Table Row
        $count = mysql_num_rows($result);

        // Result is matched username and password table row is 1
        if($count==1){

        // Register $myusername, $mypassword to redirect to file "Login_success.php
        session_start("myusername");
        session_start("mypassword");
        header("location:portal_home.php");
        }
        else{
                echo "Wrong Username or Password";
        }
?>
```

portal_home.php:

```
        <?php

        //Check if session is registered if not, redirect to main page
        session_start();
        if(!session_id()){
        header("location:main_login.php");
        }
        ?>

        <html>

        <head>
        <meta content="text/html; charset=utf-8" http-equiv="Content-Type">
        <title>Untitled 1</title>
        </head>

        <frameset rows="95,*">
        <frame name="banner" noresize="noresize" scrolling="no"
    src="raspitor_senior_project_Banner.html">
        <frameset cols="150,*">
            <frame name="contents" src="menu_bar.html">
            <frame name="main" src="Portal_Front_Page.html">
        </frameset>
        <noframes>
        <body>

        <p>This page uses frames, but your browser doesn&#39;t support them.</p>

        </body>
        </noframes>
    </frameset>

</html>
```

menu_bar.html:

```
    <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
    <html xmlns="http://www.w3.org/1999/xhtml">

    <head>
    <meta content="en-us" http-equiv="Content-Language" />
    <meta content="text/html; charset=utf-8" http-equiv="Content-Type" />
    <title>Menu Bar</title>
    </head>

    <body style="height: 250px">
    <div><a href="Home.php" target="main">Home Page</a></div>
    <hr />
    <div>
        <a href="pChart2/Temperature.php" target="main">Temperature</a></div>
    <hr />
    <div>
        <a href="pChart2/humidity.php" target="main">Humidity</a></div>
    <hr />


    </body>

</html>
```

Portal_Front_Page.html:

```
    <div>
        Welcome to the Raspitor Senior Project!<br />
```

```
                The Raspitor Project was created in an attempt to build an open source, low
                cost monitoring system for the physical environment in server rooms that
                works in combination with the incumbent network management system. The
                Project contains several sensors including temperature and humidity that
        collects real time data of the network equippment&#39;s health.</div>
```

raspitor_senior_project_Banner.html:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

<head>
<meta content="text/html; charset=utf-8" http-equiv="Content-Type" />
<title>Raspitor Senior Project</title>
<style type="text/css">
.auto-style1 {
	font-size: xx-large;
	font-family: Arial, Helvetica, sans-serif;
}
</style>
</head>

<body>

<div>
	<img alt="" height="64" src="images/Raspberrythumb.jpg" width="64" />
	<span class="auto-style1">Raspitor Senior Project</span>
	</div>

</body>

</html>
```

pchartCcompare.php:

```
<?php
// Includes Class
include("class/pDraw.class.php");
include("class/pImage.class.php");
include("class/pData.class.php");

// Create obj
$myData = new pData();

// Connect to database
$db = mysql_connect("localhost", "root", "", "raspitor");
```

```
mysql_select_db("mydatabase", $db);

// Query
$Request = "SELECT `date` , `Celcius` FROM `raspitor`.`temperature` WHERE `sensors`
= 'S1' ORDER BY `temperature`.`date` DESC LIMIT 0 , 10";
$result = mysql_query($Request, $db);

if($result === FALSE) {
        die(mysql_error());
}
while ($row = mysql_fetch_array($result))
        {
        // Gather Data from query
        $date                   = $row["date"];
        $Celcius                = $row["Celcius"];
        //Saves data
        $myData->addPoints($date, "date");
        $myData->addPoints($Celcius, "Sensor 1");
        }

// Query
$Request = "SELECT `Celcius` FROM `raspitor`.`temperature` WHERE `sensors` = 'S2'
ORDER BY `temperature`.`date` DESC LIMIT 0 , 10";
$result = mysql_query($Request, $db);

if($result === FALSE) {
        die(mysql_error());
}
while ($row = mysql_fetch_array($result))
        {
        // Gather Data from query
        $Celcius                = $row["Celcius"];
        //Saves data
        $myData->addPoints($Celcius, "Sensor 2");
        }

// put the timestamp
$myData->setAbscissa("date");

//Associate celcs on 2nd axis
$myData->setSerieOnAxis("Sensor 1",0);

// Axis name Time
$myData->setXAxisDisplay(AXIS_FORMAT_DEFAULT);

// First Y axis dedicated for Celcius
```

```
	$myData->setAxisName(0,"Sensor 1");
	$myData->setAxisUnit(0,"°C");

	//Pchart OBJ
	$myPicture = new pImage(800,300,$myData);

	//Background
	$Settings = array("R"=>166, "G"=>166, "B"=>166, "Dash"=>1, "DashR"=>186,
	"DashG"=>186, "DashB"=>186);
	$myPicture->drawFilledRectangle(0,0,800,300,$Settings);

	//Define Boundaries
	$myPicture->setGraphArea(70,70,750,200);

	//Font
	$myPicture-
	>setFontProperties(Array("fontname"=>"fonts/Forgotte.ttf","FontSize"=>11));

	// Adds Title
	$myPicture->drawText(384,25,"Temperature Sensor Celcius Comparison");
	//$TextSettings = array("Align"=>TEXT_ALIGN_TOPLEFT, "R"=>0, "G"=>0, "B"=>0);

	//Draw Scale
	$myPicture->drawScale(array("LabelRotation"=>45));

	//Draw Line Graph
	$Config = array("DisplayValues"=>1, "BreakVoid"=>0, "BreakR"=>234, "BreakG"=>55,
	"BreakB"=>26);
	$myPicture->drawLineChart($Config);

	$Config = array("FontR"=>0, "FontG"=>0, "FontB"=>0,
	"FontName"=>"fonts/Forgotte.ttf", "FontSize"=>12, "Margin"=>6, "Alpha"=>30,
	"BoxSize"=>5, "Style"=>LEGEND_NOBORDER
	, "Mode"=>LEGEND_HORIZONTAL
	);
	$myPicture->drawLegend(649,16,$Config);

	$myPicture->autoOutput("Ccompare.png");


?>
```

pchartFcompare.php:

```
	<?php
	// Includes Class
```

```
include("class/pDraw.class.php");
include("class/pImage.class.php");
include("class/pData.class.php");

// Create obj
$myData = new pData();

// Connect to database
$db = mysql_connect("localhost", "root", "", "raspitor");
mysql_select_db("mydatabase", $db);

// Query
$Request = "SELECT `date` , `fahrenheit` FROM `raspitor`.`temperature` WHERE
`sensors` = 'S1' ORDER BY `temperature`.`date` DESC LIMIT 0 , 10";
$result = mysql_query($Request, $db);

if($result === FALSE) {
	die(mysql_error());
}
while ($row = mysql_fetch_array($result))
	{
	// Gather Data from query
	$date			= $row["date"];
	$fahrenheit	= $row["fahrenheit"];
	//Saves data
	$myData->addPoints($date, "date");
	$myData->addPoints($fahrenheit, "Sensor 1");
	}

//Saves data
//$myData->addPoints($date, "date");
//$myData->addPoints($fahrenheit, "fahrenheit");

// Query
$Request = "SELECT `date` , `fahrenheit` FROM `raspitor`.`temperature` WHERE
`sensors` = 'S2' ORDER BY `temperature`.`date` DESC LIMIT 0 , 10";
$result = mysql_query($Request, $db);

if($result === FALSE) {
	die(mysql_error());
}
while ($row = mysql_fetch_array($result))
	{
	// Gather Data from query
	$fahrenheit	= $row["fahrenheit"];
	//Saves data
```

```
        $myData->addPoints($fahrenheit, "Sensor2");
        }

// put the timestamp
$myData->setAbscissa("date");

//Associate Sensors on 2nd axis
$myData->setSerieOnAxis("Sensor 1",0);


// Axis name Time
$myData->setXAxisDisplay(AXIS_FORMAT_DEFAULT);

// First Y axis dedicated for Celcius
$myData->setAxisName(0,"Temperature");
$myData->setAxisUnit(0,"°F");

//Pchart OBJ
$myPicture = new pImage(800,300,$myData);

//Background
$Settings = array("R"=>166, "G"=>166, "B"=>166, "Dash"=>1, "DashR"=>186,
"DashG"=>186, "DashB"=>186);
$myPicture->drawFilledRectangle(0,0,800,300,$Settings);

//Define Boundaries
$myPicture->setGraphArea(70,70,750,200);

//Font
$myPicture-
>setFontProperties(Array("fontname"=>"fonts/Forgotte.ttf","FontSize"=>11));

// Adds Title
$myPicture->drawText(384,25,"Temperature Sensor Fahrenheit Comparison");

//Draw Scale
$myPicture->drawScale(array("LabelRotation"=>45));

//Draw Line graph
$Config = array("DisplayValues"=>1, "BreakVoid"=>0, "BreakR"=>234, "BreakG"=>55,
"BreakB"=>26);
$myPicture->drawLineChart($Config);

$Config = array("FontR"=>0, "FontG"=>0, "FontB"=>0,
"FontName"=>"fonts/Forgotte.ttf", "FontSize"=>12, "Margin"=>6, "Alpha"=>30,
"BoxSize"=>5, "Style"=>LEGEND_NOBORDER
```

```
	, "Mode"=>LEGEND_HORIZONTAL
	);
	$myPicture->drawLegend(649,16,$Config);

	$myPicture->autoOutput("Fcompare.png");
?>
```

PchartHumidity.php:

```
	<?php
	// Includes Class
	include("class/pDraw.class.php");
	include("class/pImage.class.php");
	include("class/pData.class.php");

	// Create obj
	$myData = new pData();

	// Connect to database
	$db = mysql_connect("localhost", "root", "", "raspitor");
	mysql_select_db("mydatabase", $db);

	// Query
	$Request = "SELECT `date` , `humidity` FROM `raspitor`.`humidity` WHERE `sensors`
	= 'H1' && `humidity` != 99.9 && `humidity` != 32.0 ORDER BY `humidity`.`date`
	DESC LIMIT 0 , 10";
	$result = mysql_query($Request, $db);

	if($result === FALSE) {
		die(mysql_error());
	}
	while ($row = mysql_fetch_array($result))
		{
		// Gather Data from query
		$date[]			= $row["date"];
		$humidity[]	= $row["humidity"];
		}

	//Saves data
	$myData->addPoints($date, "date");
	$myData->addPoints($humidity, "humidity");

	// put the timestamp
	$myData->setAbscissa("date");

	//Associate celcs on 2nd axis
	$myData->setSerieOnAxis("humidity",0);
```

```
        // Axis name Time
        $myData->setXAxisDisplay(AXIS_FORMAT_DEFAULT);

        // First Y axis dedicated for Celcius
        $myData->setAxisName(0,"Humidity");
        $myData->setAxisUnit(0,"%");

        //Pchart OBJ
        $myPicture = new pImage(800,300,$myData);

        //Background
        $Settings = array("R"=>166, "G"=>166, "B"=>166, "Dash"=>1, "DashR"=>186,
        "DashG"=>186, "DashB"=>186);
        $myPicture->drawFilledRectangle(0,0,800,300,$Settings);

        //Define Boundaries
        $myPicture->setGraphArea(70,70,750,200);

        //Font
        $myPicture-
        >setFontProperties(Array("fontname"=>"fonts/Forgotte.ttf","FontSize"=>11));

        // Adds Title
        $myPicture->drawText(384,25,"Humidity Sensor");

        //Draw Scale
        $myPicture->drawScale(array("LabelRotation"=>45));

        //Draw Line graph
        $Config = array("DisplayValues"=>1, "BreakVoid"=>0, "BreakR"=>234, "BreakG"=>55,
        "BreakB"=>26);
        $myPicture->drawLineChart($Config);

        $Config = array("FontR"=>0, "FontG"=>0, "FontB"=>0,
        "FontName"=>"fonts/Forgotte.ttf", "FontSize"=>12, "Margin"=>6, "Alpha"=>30,
        "BoxSize"=>5, "Style"=>LEGEND_NOBORDER
        , "Mode"=>LEGEND_HORIZONTAL
        );
        $myPicture->drawLegend(649,16,$Config);

        $myPicture->autoOutput("humid.png");

?>
```

pchartS1C.php

```
        <?php
```

```
// Includes Class
include("class/pDraw.class.php");
include("class/pImage.class.php");
include("class/pData.class.php");

// Create obj
$myData = new pData();

// Connect to database
$db = mysql_connect("localhost", "root", "", "raspitor");
mysql_select_db("mydatabase", $db);

// Query
$Request = "SELECT `date` , `Celcius` FROM `raspitor`.`temperature` WHERE `sensors`
= 'S1' ORDER BY `temperature`.`date` DESC LIMIT 0 , 10";
$result = mysql_query($Request, $db);

if($result === FALSE) {
        die(mysql_error());
}
while ($row = mysql_fetch_array($result))
        {
        // Gather Data from query
        $date[]                 = $row["date"];
        $Celcius[]              = $row["Celcius"];
        }

//Saves data
$myData->addPoints($date, "date");
$myData->addPoints($Celcius, "Celcius");

// put the timestamp
$myData->setAbscissa("date");

//Associate celcs on 2nd axis
$myData->setSerieOnAxis("Celcius",0);

// Axis name Time
$myData->setXAxisDisplay(AXIS_FORMAT_DEFAULT);

// First Y axis dedicated for Celcius
$myData->setAxisName(0,"Celcius");
$myData->setAxisUnit(0,"°C");

//Pchart OBJ
$myPicture = new pImage(800,300,$myData);
```

```
//Background
$Settings = array("R"=>166, "G"=>166, "B"=>166, "Dash"=>1, "DashR"=>186,
"DashG"=>186, "DashB"=>186);
$myPicture->drawFilledRectangle(0,0,800,300,$Settings);

//Define Boundaries
$myPicture->setGraphArea(70,70,750,200);

//Font
$myPicture-
>setFontProperties(Array("fontname"=>"fonts/Forgotte.ttf","FontSize"=>11));

// Adds Title
$myPicture->drawText(384,25,"Temperature Sensor S1 Celcius");
//$TextSettings = array("Align"=>TEXT_ALIGN_TOPLEFT, "R"=>0, "G"=>0, "B"=>0);

//Draw Scale
$myPicture->drawScale(array("LabelRotation"=>45));

//Draw Line Graph
$Config = array("DisplayValues"=>1, "BreakVoid"=>0, "BreakR"=>234, "BreakG"=>55,
"BreakB"=>26);
$myPicture->drawLineChart($Config);

$Config = array("FontR"=>0, "FontG"=>0, "FontB"=>0,
"FontName"=>"fonts/Forgotte.ttf", "FontSize"=>12, "Margin"=>6, "Alpha"=>30,
"BoxSize"=>5, "Style"=>LEGEND_NOBORDER
, "Mode"=>LEGEND_HORIZONTAL
);
$myPicture->drawLegend(649,16,$Config);

$myPicture->autoOutput("S2C.png");

?>
```

pchartS1F.php:

```
<?php
// Includes Class
include("class/pDraw.class.php");
include("class/pImage.class.php");
include("class/pData.class.php");

// Create obj
$myData = new pData();

// Connect to database
```

```
$db = mysql_connect("localhost", "root", "", "raspitor");
mysql_select_db("mydatabase", $db);

// Query
$Request = "SELECT `date` , `fahrenheit` FROM `raspitor`.`temperature` WHERE
`sensors` = 'S1' ORDER BY `temperature`.`date` DESC LIMIT 0 , 10";
$result = mysql_query($Request, $db);

if($result === FALSE) {
        die(mysql_error());
}
while ($row = mysql_fetch_array($result))
        {
        // Gather Data from query
        $date[]                 = $row["date"];
        $fahrenheit[]   = $row["fahrenheit"];
        }

//Saves data
$myData->addPoints($date, "date");
$myData->addPoints($fahrenheit, "fahrenheit");

// put the timestamp
$myData->setAbscissa("date");

//Associate celcs on 2nd axis
$myData->setSerieOnAxis("fahrenheit",0);

// Axis name Time
$myData->setXAxisDisplay(AXIS_FORMAT_DEFAULT);

// First Y axis dedicated for Celcius
$myData->setAxisName(0,"Temperature");
$myData->setAxisUnit(0,"°F");

//Pchart OBJ
$myPicture = new pImage(800,300,$myData);

//Background
$Settings = array("R"=>166, "G"=>166, "B"=>166, "Dash"=>1, "DashR"=>186,
"DashG"=>186, "DashB"=>186);
$myPicture->drawFilledRectangle(0,0,800,300,$Settings);

//Define Boundaries
$myPicture->setGraphArea(70,70,750,200);
```

```
    //Font
    $myPicture-
    >setFontProperties(Array("fontname"=>"fonts/Forgotte.ttf","FontSize"=>11));

    // Adds Title
    $myPicture->drawText(384,25,"Temperature Sensor S1 fahrenheit");

    //Draw Scale
    $myPicture->drawScale(array("LabelRotation"=>45));

    //Draw Line graph
    $Config = array("DisplayValues"=>1, "BreakVoid"=>0, "BreakR"=>234, "BreakG"=>55,
    "BreakB"=>26);
    $myPicture->drawLineChart($Config);

    $Config = array("FontR"=>0, "FontG"=>0, "FontB"=>0,
    "FontName"=>"fonts/Forgotte.ttf", "FontSize"=>12, "Margin"=>6, "Alpha"=>30,
    "BoxSize"=>5, "Style"=>LEGEND_NOBORDER
    , "Mode"=>LEGEND_HORIZONTAL
    );
    $myPicture->drawLegend(649,16,$Config);

    $myPicture->autoOutput("S1F.png");
?>
```

pchartS2C.php:

```
    <?php
    // Includes Class
    include("class/pDraw.class.php");
    include("class/pImage.class.php");
    include("class/pData.class.php");

    // Create obj
    $myData = new pData();

    // Connect to database
    $db = mysql_connect("localhost", "root", "", "raspitor");
    mysql_select_db("mydatabase", $db);

    // Query
    $Request = "SELECT `date` , `Celcius` FROM `raspitor`.`temperature` WHERE `sensors`
    = 'S1' ORDER BY `temperature`.`date` DESC LIMIT 0 , 10";
    $result = mysql_query($Request, $db);

    if($result === FALSE) {
        die(mysql_error());
```

```
}
while ($row = mysql_fetch_array($result))
        {
        // Gather Data from query
        $date[]                 = $row["date"];
        $Celcius[]              = $row["Celcius"];
        }

//Saves data
$myData->addPoints($date, "date");
$myData->addPoints($Celcius, "Celcius");

// put the timestamp
$myData->setAbscissa("date");

//Associate celcs on 2nd axis
$myData->setSerieOnAxis("Celcius",0);

// Axis name Time
$myData->setXAxisDisplay(AXIS_FORMAT_DEFAULT);

// First Y axis dedicated for Celcius
$myData->setAxisName(0,"Celcius");
$myData->setAxisUnit(0,"°C");

//Pchart OBJ
$myPicture = new pImage(800,300,$myData);

//Background
$Settings = array("R"=>166, "G"=>166, "B"=>166, "Dash"=>1, "DashR"=>186,
"DashG"=>186, "DashB"=>186);
$myPicture->drawFilledRectangle(0,0,800,300,$Settings);

//Define Boundaries
$myPicture->setGraphArea(70,70,750,200);

//Font
$myPicture-
>setFontProperties(Array("fontname"=>"fonts/Forgotte.ttf","FontSize"=>11));

// Adds Title
$myPicture->drawText(384,25,"Temperature Sensor S2 Celcius");
//$TextSettings = array("Align"=>TEXT_ALIGN_TOPLEFT, "R"=>0, "G"=>0, "B"=>0);

//Draw Scale
$myPicture->drawScale(array("LabelRotation"=>45));
```

```
	//Draw Line Graph
	$Config = array("DisplayValues"=>1, "BreakVoid"=>0, "BreakR"=>234, "BreakG"=>55,
	"BreakB"=>26);
	$myPicture->drawLineChart($Config);

	$Config = array("FontR"=>0, "FontG"=>0, "FontB"=>0,
	"FontName"=>"fonts/Forgotte.ttf", "FontSize"=>12, "Margin"=>6, "Alpha"=>30,
	"BoxSize"=>5, "Style"=>LEGEND_NOBORDER
	, "Mode"=>LEGEND_HORIZONTAL
	);
	$myPicture->drawLegend(649,16,$Config);

	$myPicture->autoOutput("S2C.png");

?>
```

pchartS2F.php:

```
	<?php
	// Includes Class
	include("class/pDraw.class.php");
	include("class/pImage.class.php");
	include("class/pData.class.php");

	// Create obj
	$myData = new pData();

	// Connect to database
	$db = mysql_connect("localhost", "root", "", "raspitor");
	mysql_select_db("mydatabase", $db);

	// Query
	$Request = "SELECT `date` , `fahrenheit` FROM `raspitor`.`temperature` WHERE
	`sensors` = 'S2' ORDER BY `temperature`.`date` DESC LIMIT 0 , 10";
	$result = mysql_query($Request, $db);

	if($result === FALSE) {
		die(mysql_error());
	}
	while ($row = mysql_fetch_array($result))
		{
		// Gather Data from query
		$date[]			= $row["date"];
		$fahrenheit[]	= $row["fahrenheit"];
		}
```

```
//Saves data
$myData->addPoints($date, "date");
$myData->addPoints($fahrenheit, "fahrenheit");

// put the timestamp
$myData->setAbscissa("date");

//Associate celcs on 2nd axis
$myData->setSerieOnAxis("fahrenheit",0);

// Axis name Time
$myData->setXAxisDisplay(AXIS_FORMAT_DEFAULT);

// First Y axis dedicated for Celcius
$myData->setAxisName(0,"Temperature");
$myData->setAxisUnit(0,"°F");

//Pchart OBJ
$myPicture = new pImage(800,300,$myData);

//Background
$Settings = array("R"=>166, "G"=>166, "B"=>166, "Dash"=>1, "DashR"=>186,
"DashG"=>186, "DashB"=>186);
$myPicture->drawFilledRectangle(0,0,800,300,$Settings);

//Define Boundaries
$myPicture->setGraphArea(70,70,750,200);

//Font
$myPicture-
>setFontProperties(Array("fontname"=>"fonts/Forgotte.ttf","FontSize"=>11));

// Adds Title
$myPicture->drawText(384,25,"Temperature Sensor S2 fahrenheit");

//Draw Scale
$myPicture->drawScale(array("LabelRotation"=>45));

//Draw Line graph
$Config = array("DisplayValues"=>1, "BreakVoid"=>0, "BreakR"=>234, "BreakG"=>55,
"BreakB"=>26);
$myPicture->drawLineChart($Config);

$Config = array("FontR"=>0, "FontG"=>0, "FontB"=>0,
"FontName"=>"fonts/Forgotte.ttf", "FontSize"=>12, "Margin"=>6, "Alpha"=>30,
"BoxSize"=>5, "Style"=>LEGEND_NOBORDER
```

```
        , "Mode"=>LEGEND_HORIZONTAL
        );
        $myPicture->drawLegend(649,16,$Config);

        $myPicture->autoOutput("S2F.png");

?>
```