

California State University, Fresno
Electrical and Computer Engineering Department

Experiment Title: Project Proposal

Course: ECE 186A: Senior Design I

Instructor: Dr. Raeisi

Prepared by: Nick Morley
Jonathan Richards

Date _____ March 9, 2016

To be filled out by the instructor

Comments: _____

Final Grade:

TABLE OF CONTENTS

Section	Page
Title Page	1
Table of Contents	2
1 The Team	3
2 Goals	3
3 Objectives	3
3.1 Modularity	3
3.2 Dynamic	3
3.3 Interactive	3
3.4 Easy to Use	3
4 Background	4
4.1 Microcontroller Programming and Interfacing	4
4.2 Wireless Communication	4
4.3 Data Storage, Processing, and Serving	4
4.4 Mobile App Development	4
5 Feasibility	4
5.1 ESP8266 Microcontroller	4
5.2 ATmega328P Microcontroller	5
5.3 2.8" TFT LCD Touchscreen	5
5.4 DHT22 Sensor	5
5.5 Uxcell AC Current Sensor	5
5.6 Barometric Pressure Sensor	6
5.7 Brightness Sensor	6
5.8 Sound Sensor	6
5.9 Cloud Server	6
6 Protocols	6
6.1 Intranode Protocol	6
6.2 DHT22 Protocol	7
7 Input/Output Diagram	11
8 Project Timeline	14

1 THE TEAM

Team Name: Sensor Network

Team Members: Nick Morley
Jonathan Richards

Technical Advisor: Dr. Hovannes Kulhandjian

Course Instructor: Dr. Reza Raeisi

2 GOALS

- Create a system to collect, transport, analyze, store, and visualize sensor data
- Control external systems manually or automatically according to sensor inputs
- Access user interface anywhere with an internet connection

3 OBJECTIVES

3.1 Modularity

- The network shall be modular and be composed of a wireless network of similar nodes with one base station
- The nodes shall be modular and support a variety of sensors and controllers interfacing over a common hardware/software interface

3.2 Dynamic

- The network shall be able to respond to the dynamic addition or removal of nodes
- The nodes shall support hot-plugging of sensors
- The client application shall view live data from the sensors

3.3 Interactive

- The nodes shall generate alerts for the user to respond to
- The user shall control the node's sensors and controllers from the client application

3.4 Easy to Use

- The user application shall allow easy configuration of new nodes and sensors
- The historic data shall be presented in a clean, graphical manner to the user
- The user interface shall be intuitive and user friendly

4 BACKGROUND

4.1 Microcontroller Programming and Interfacing

Both team members of our team have experience programming a wide variety of microcontrollers to interface with the outside world. This knowledge will be vital to interfacing with the sensors to gather data into our network and controlling exterior devices. It will also be useful for interprocessor communication to relay messages from one end of the network to the other (including the user's phone).

4.2 Wireless Communication

Nick has some experience with establishing communication between a few processors through multiple point to point links. This will help us get starting, but we will put further research into how to implement a mesh network, including routing algorithms.

4.3 Data Storage, Processing, and Serving

Both team members have some server-side experience in spinning up servers and presenting data to the user from a database. Nick also has experience funneling data into the database that will be important for establishing communication between the server and base station to transfer data into the database. Since our sensor data will be highly compressable, we will implement a custom compression algorithm to minimize network and server resources.

4.4 Mobile App Development

Both team members have some web application experience we will leverage for first creating a web interface to our network. We will then port our web application to Android and iOS.

5 FEASIBILITY

5.1 ESP8266 Microcontroller

The ESP8266 microcontroller is a low cost microcontroller with integrated, long-range wireless communications that will be the master control unit of each node. The most important features for our purposes are:

- Unit cost of \$2.44 in lots of 10 or more
- Range of 366 meters with standard consumer wireless router and range of 3.71 kilometer with long range access point
- I²C interface for communicating with sensor modules
- Analog to digital converter for monitoring supply voltage (if we deploy battery powered nodes)
- UART port (if we interface multiple ESP8266s on a node)
- 13 general purpose input/output pins for buttons and status LEDs

5.2 ATmega328P Microcontroller

The ATmega328P microcontroller is a low cost, general purpose microcontroller. To simplify the software and hardware interface between the sensors and our network, each sensor module will contain an ATmega328P. This enables a simple, common interface between the node and sensor module over I²C and offloads the sensor monitoring from the ESP8266. It also makes the sensor modules more universal since adding or changing sensors will not require a firmware update on the node itself. Instead the sensor specific firmware is stored on the ATmega328P that is bundled with the sensor. The most important features for our purposes are:

- Unit cost of \$1.21 in lots of 10 or more
- Wide variety of interfaces to talk to the sensor based on it's specific needs
 - 20 general purpose input/output pins
 - 8 channel analog to digital converter
 - 6 pulse width modulation outputs
 - SPI
 - UART
 - I²C

5.3 2.8" TFT LCD Touchscreen

The touchscreen allows the node to have a direct and convenient user interface. For example, on a thermostat it would allow the user to see the current temperature, change the set point, view the energy usage of the air conditioner, and view a plot of the temperature without having to pull up the client app. It can also be used to aid in the initial setup process. It is not strictly necessary, but offers a polished feel to the system for the user to immediately see the system working. The most important features for our purposes are:

- Unit cost of \$8.19
- Full color screen for displaying user interface
- Touch enabled for user input

5.4 DHT22 Sensor

The DHT22 temperature and humidity sensor can serve for either a weather station or thermostat node. It interfaces with the ATmega328P over a digital bus. The most important features for our purposes are:

- Unit cost of \$5.99
- Temperature range of -40 - 80 °C
- Temperature accuracy of ± 0.5 °C
- Humidity range of 0 - 100% RH
- Humidity accuracy of $\pm 2\%$ RH

5.5 AC Current Sensor

The current sensor is non-invasive and plugs into a 3.5 mm TRS connector. Since it is non-invasive, the client safely clips it around the device they would like to measure without

having to cut any wires. The most important features for our purposes are:

- Unit cost of \$11.19
- Measures up to 100 A loads (can be used for air conditioners or entire house)
- Common connector

5.6 Barometric Pressure Sensor

The barometric pressure sensor allows us to enhance a weather station's measurements or track the node's elevation. The most important features for our purposes are:

- Unit cost of \$4.34
- Pressure range of 300 - 1100 hPa
- Pressure accuracy of 0.02 hPa
- Elevation range of -500 - 9000 meters
- Elevation accuracy of 17 cm
- I²C interface

5.7 Brightness Sensor

The brightness sensor is a photoresistor that can detect the ambient brightness level of the room. It can be used to monitor when people leave the lights on or track sunrise/sunset. We can also mount multiple sensors on each module pointed in different directions. The most important features for our purposes are:

- Unit cost of \$0.03 in lots of 100 or more

5.8 Sound Sensor

The sound sensor is a microphone that can detect people talking or serve as a spectrum analyzer using digital signal processing.

5.9 Cloud Server

The cloud server will store historic sensor data and serve the web application for the users in addition to hosting the project management software for this project. It is the single entry point for users to access the sensor network from the internet.

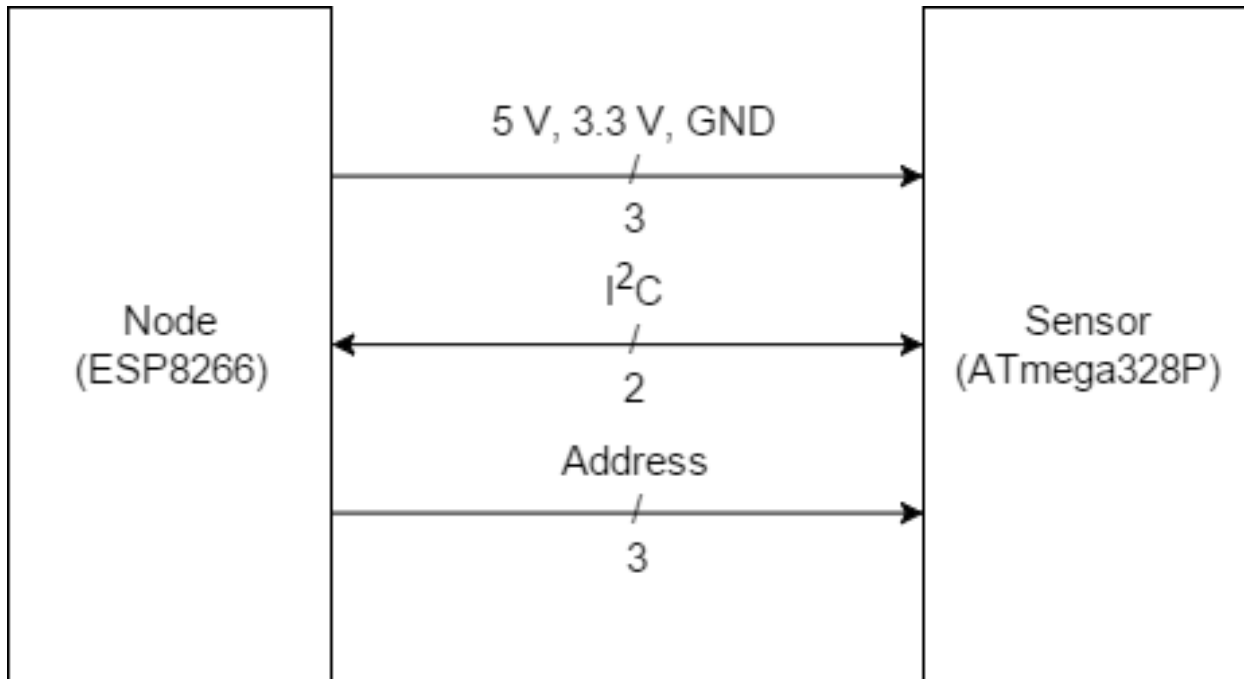
6 PROTOCOLS

6.1 Intranode Protocol

The Intranode Protocol is used to gather sensor data from the sensor and controller modules to the node. The physical layer diagramed in Figure 6.1 depicts pins connecting the systems. The link layer is composed of 32 byte packets exchanged between the systems as described in Table 6.1. All words are stored in little-endian format.

Table 6.1: Intranode Packets

Field Name	Byte Index	Field Width	Field Explan
Address	0	8 bits	Which port the sensor is plugged into
Message Type	1:2	16 bits	An identifier for how the payload is formatted
Sensor Type	3:4	16 bits	An identifier for the physical sensor board
Reserved	5:6	16 bits	Reserved for later revisions
Payload	7:30	24 bytes	Message data
Checksum	31	8 bits	Checksum of first 31 bytes of packet

**Figure 6.1:** The connections between the node and sensor module

6.2 DHT22 Protocol

The DHT22 communicates over a digital, bi-directional, single pin bus that is normally high. To begin a transmission, the ATmega first brings the bus low for at least 1 ms by configuring the pin as a digital output with a value of zero. It then configures the pin as a digital input with a weak pullup. 40 μ s later, the DHT22 brings the bus low for 80 μ s and high for 80 μ s to indicate that it is about to begin transmission as shown in Figure 6.2. The DHT22 then brings the bus low for 50 μ s and high for a variable amount of time. If the bus is high for 26 - 28 μ s, the bit is interpreted as a “0” as shown in Figure 6.3. If the bus is high for 70 μ s, the bit is interpreted as a “1” as shown in Figure 6.4. After all 40 bits have been transmitted, the DHT22 lets the bus float high. Note that the highest bit is transmitted first.

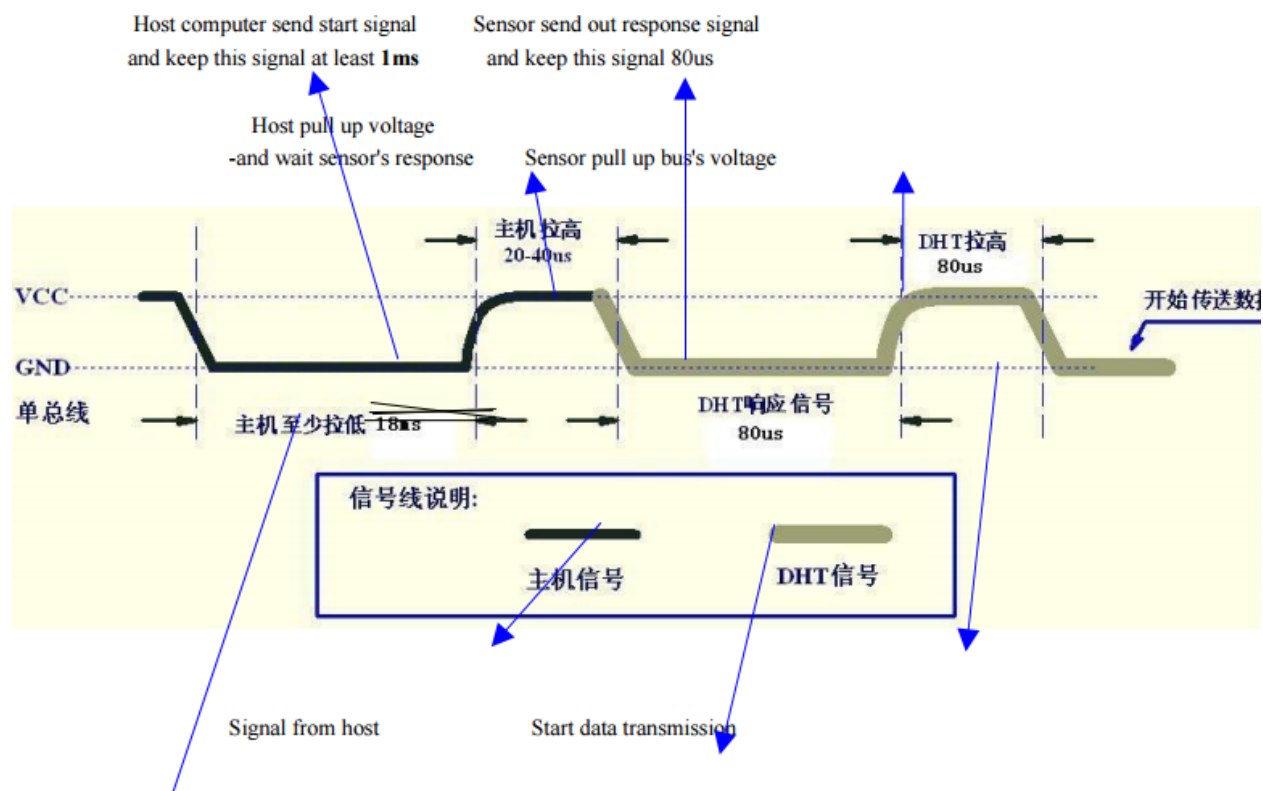


Figure 6.2: The initialization of a reading between the ATmega328P and DHT22

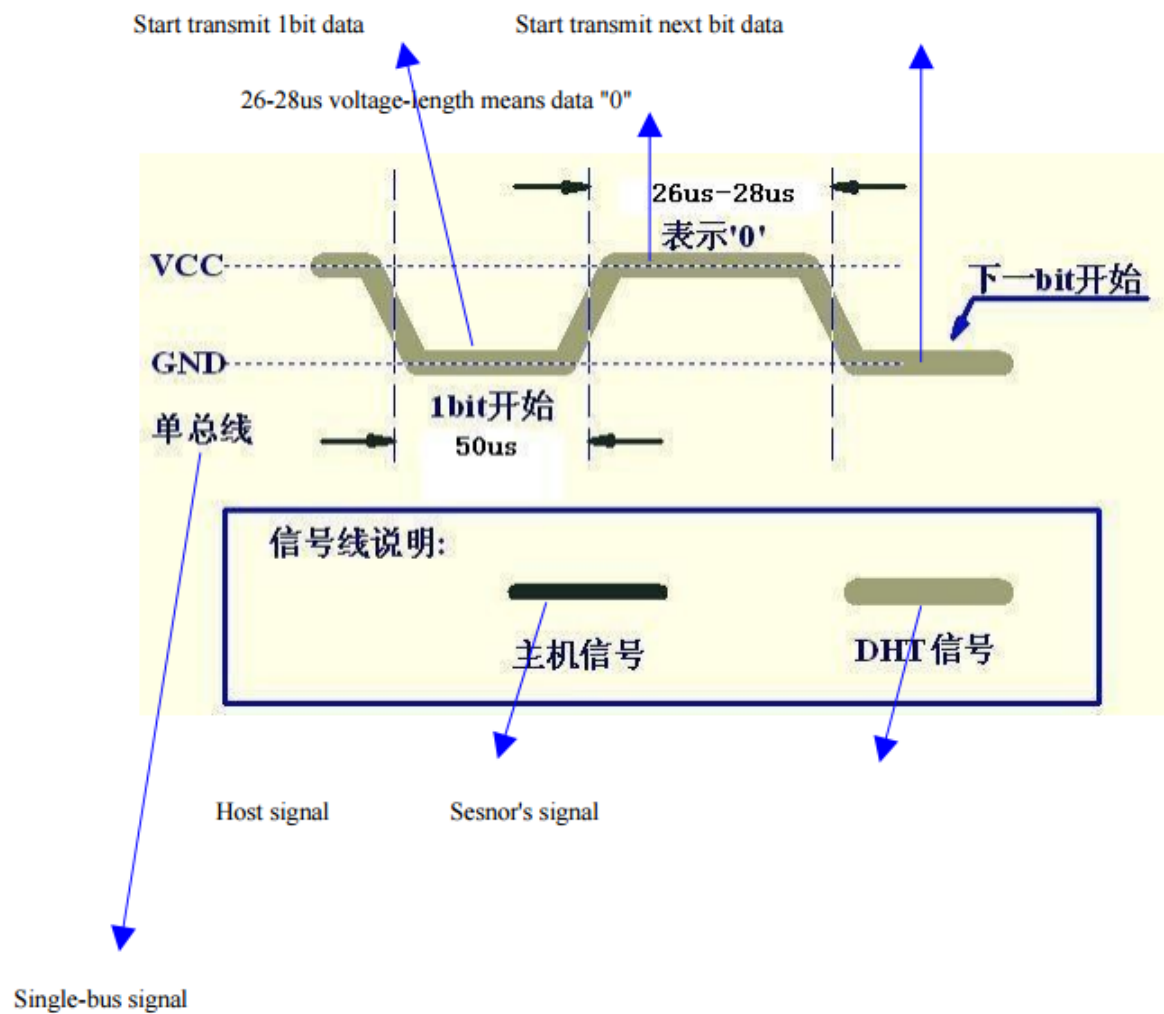


Figure 6.3: The DHT22 trasnmitting a "0"

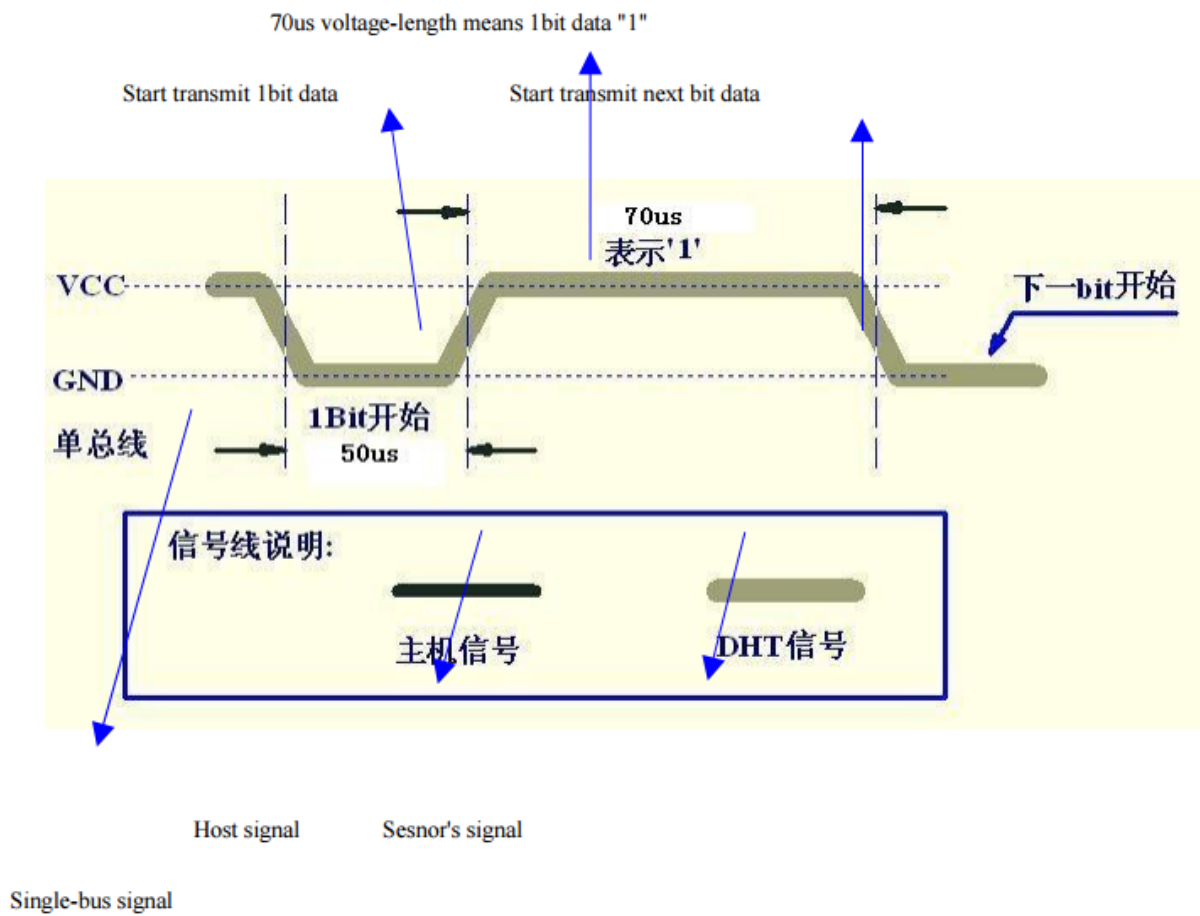


Figure 6.4: The DHT22 transmitting a "1"

6.3 AC Current Sensor

The current sensor uses a transformer to induce a current between 0 and 50 mA (RMS) at 60 Hz on the secondary coil. This wave is sampled on the ATmega328P's analog input channel to calculate the RMS current through the primary coil. With the 10 bit ADC on the ATmega328P, this translates to a 98 mA (RMS) resolution on the primary coil's current.

6.4 Barometric Pressure Sensor

The barometric pressure sensor communicates with the ATmega328P through an I²C bus.

6.5 Brightness Sensor

The brightness sensors connect to the ATmega328P's ADC channels where a higher analog voltage represents a brighter room.

6.6 Sound Sensor

The sound sensor

7 INPUT/OUTPUT DIAGRAM

The logical datapath for the project is diagramed in Figure 7.1. The network diagram for connecting different physical modules are presented in Figure 7.2. The inputs/outputs are summarized below.

- Sensor Data
 - Temperature
 - Humidity
 - Barometric Pressure / Elevation
 - Current / Energy usage
 - Brightness
 - Sound
- Control Devices
 - Air Conditioner / Heater
 - LEDs
 - Outlets
 - Lights
- Node Configuration
 - Sampling Rate
 - Precision
 - Control Conditions
 - Network
 - Link to Account
- Plotted Sensor Data
 - Organize sensor data graphically for user
- Sensor Data Analytics

- Averages
- Cumulative Totals
- Network Status
 - Nodes online
 - Last sensor readings

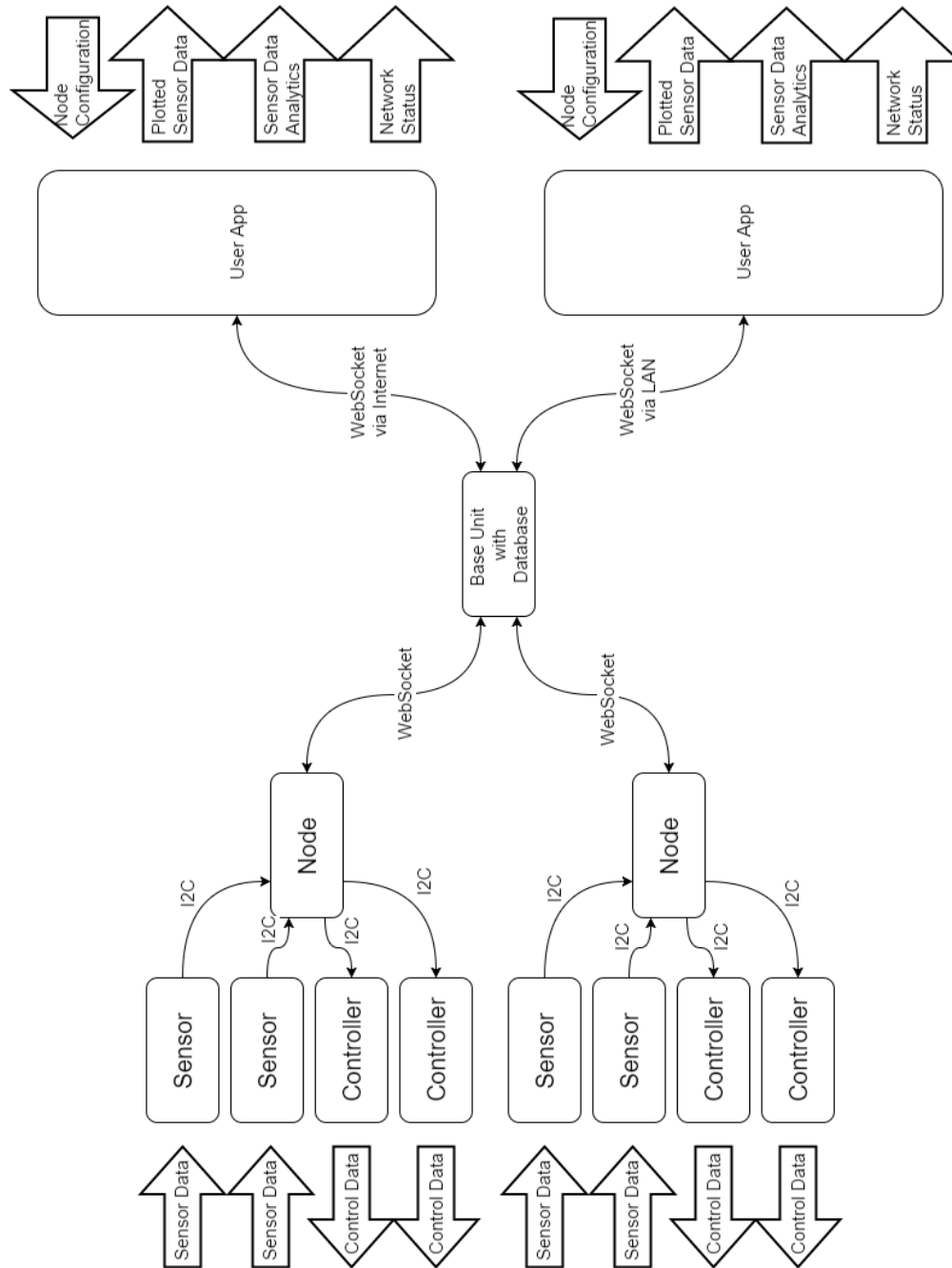


Figure 7.1: The Logical Datapath

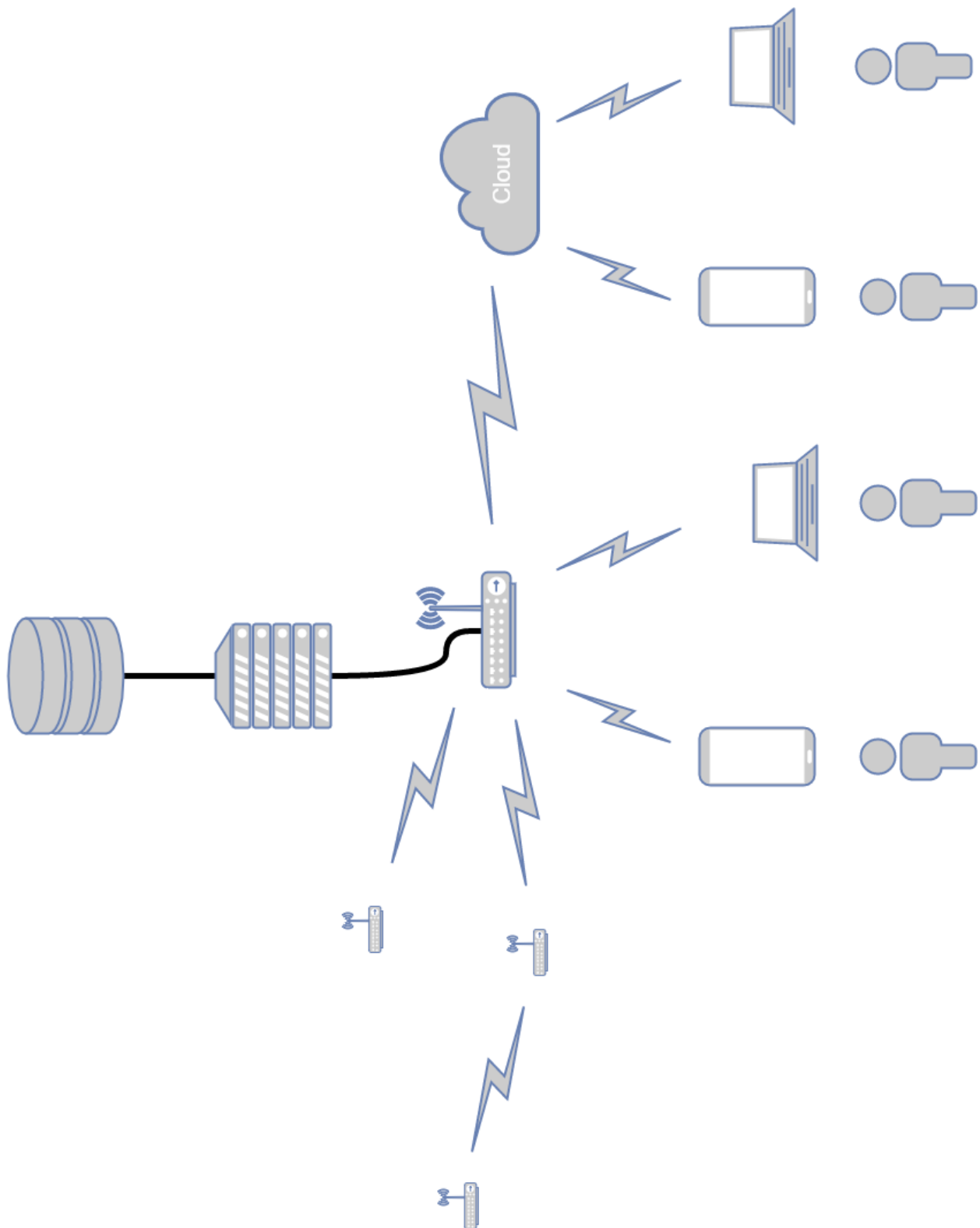


Figure 7.2: The Physical Datapath

8 PROJECT TIMELINE

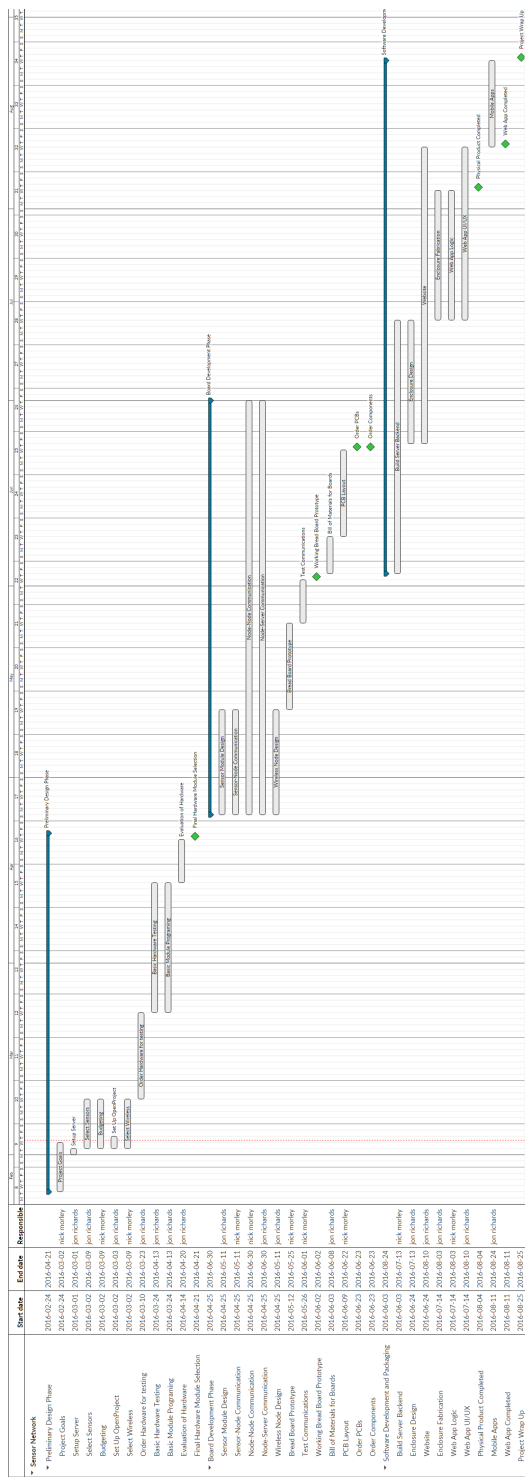


Figure 8.1: Gantt Chart for Project Timeline