### GAU GAS METERING AND BILLING AUTOMATION

### Software Requirements Specification

### <<Version 1.0>>

### Participants

**Vinay Kiran M**

**Perumal S**

**Tilak K**

**Vinay Bhat**

Contents

Description 3

Product Functions 3

User Characteristics 4

Constraints 4

Assumptions and Dependencies 4

External Resource Requirements 4

User Interface 4

Hardware Interface 5

Software Interface 5

Communications Interface 5

Functional Requirements Views 5

Primary List of Functional Requirements 6

<Req\_1> 6

<Req\_2> 10

<Req\_3> 10

<Req\_4> 11

<Req\_5> 11

<Req\_6> 11

<Req\_7> 11

<Req\_8>………………………………………………………………………….11

Functional Requirements 17

Non-Functional Requirements 17

Availability 18

Security 18

Maintainability 18

**DESCRIPTION**

Sustain Earth uses innovative Biogas technology to provide affordable, clean cooking gas rural communities .By rebranding the now-notorious Biogas product as Gau-Gas and implementing new materials, new processes, and new technology. SustainEarth solves the problems faced by the last generation of Biogas users and change the perception about the usefulness of Biogas.

* **Gau-Gas** systems are easier to install, more efficient, easier to maintain.
* They use Internet and Mobile connectivity to collect the usage logs.
* Since these systems are present in rural areas ,we cannot defy an internet solution.
* The main disadvantage is the manual collection of usage logs.
* Based on these usage logs billing is done.
* Main challenge is to reduce manual intervention and to fully automate the billing and metering of those systems.
* And to devise a no Internet and easy usable solution since digital literacy of rural areas is less.
* A list of the requirements presented numerically.

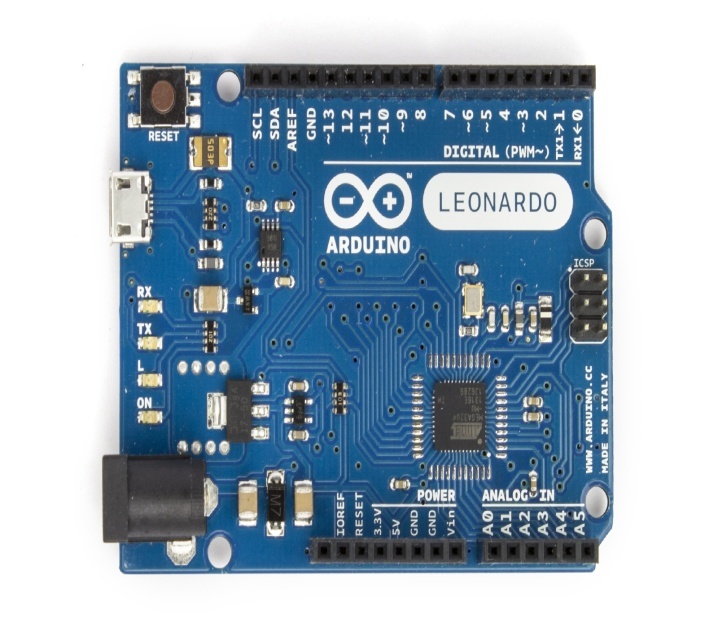
The document is intended to establish the initial scope of the development effort.

.

**PRODUCT FUNCTIONS:**

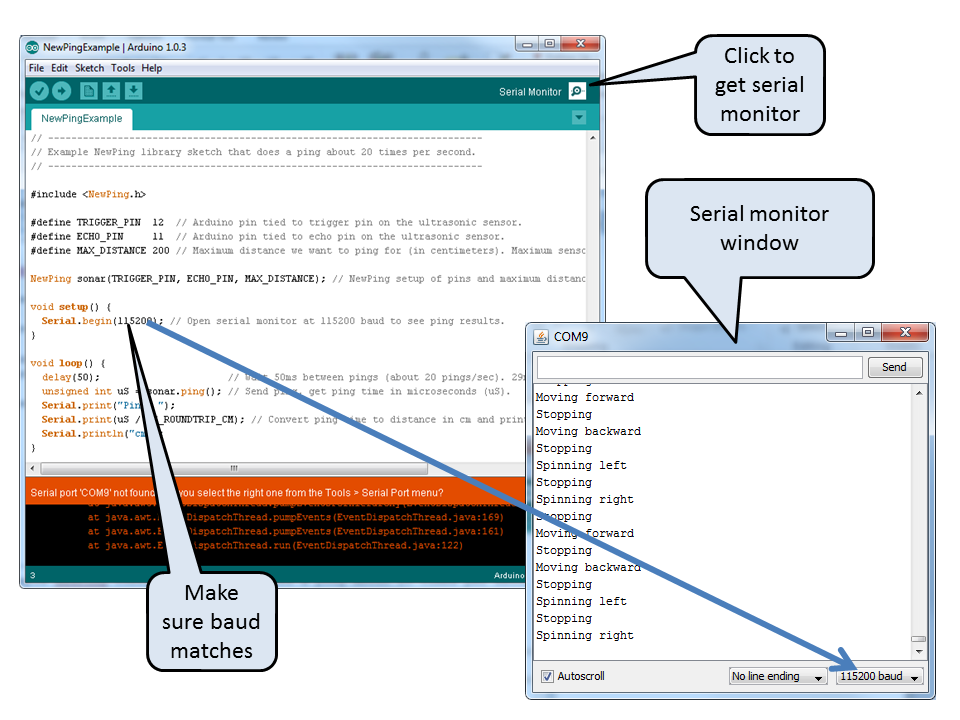
We select the Arduino Leonardo board to calculate the total run time of the pump . The Arduino Leonardo is a microcontroller board based on the ATmega32u4 . It has 20 digital input/output pins (of which 7 can be used as PWM outputs and 12 as analog inputs), a 16 MHz crystal oscillator, a micro USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

The Leonardo differs from all preceding boards in that the ATmega32u4 has built-in USB communication, eliminating the need for a secondary processor. This allows the Leonardo to appear to a connected computer as a mouse and keyboard, in addition to a virtual (CDC) serial / COM port.



The programming work is simpler by the usage of micro USB cable that connects to the computer or laptop having the required software and hardware specifications . The Arduino Leonardo board is used to calculate the Pump ON time and OFF time . However we only require the ON time since that is the time we need to calculate the USAGE LOGISTICS of the GAU GAS consumed by the customer .

Arduino Leonardo has an inbuilt millis() function which return the number of milliseconds since the Arduino board began running the current program . This function returns the values on a separate window called as SERIAL MONITOR with the specific frequency ,i.e. baud rate mentioned .



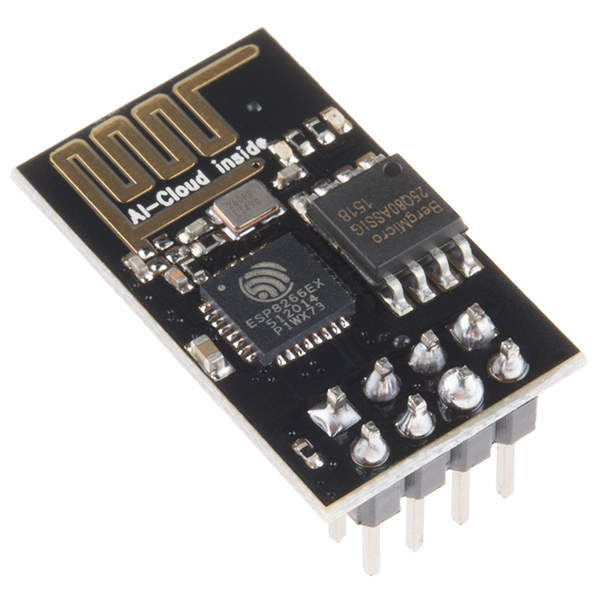
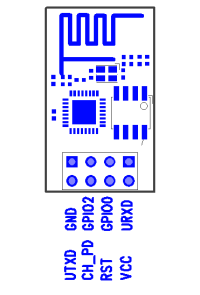
**Constraint of Arduino Leonardo:**

Arduino is controller dependent and its hardware capability keeps on changing .It communicates with the help of USB to serial converter and every time the Leonardo is reset it will create a new virtual COM port with a different number.

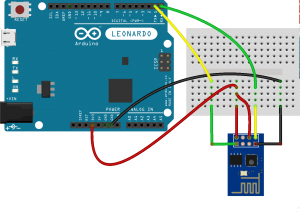
So , the usage logistics of the gas cannot be retrieved when the device gets reset .To , overcome this drawback of Leonardo we use the ESP8266 WiFi module .

**Arduino Leanordo with ESP8266 module:**

ESP8266 has 8 pins, 4 in the row of 2. The first pin on the top left is GND. The two pins right from the GND are GPIO 2 and 0. The pin on the top right side is the RX pin and the pin on the lower left is TX. These are the pins for communication. The middle pins on the bottom are CH\_PD(chip power-down) and RST(reset).

The main advantage of ESP8266 module is, coding can be done by using same ARDUINO IDE software and it also has inbuilt EEPROM memory. The connections to the Arduino Leonardo are shown below as in the diagram .



NOTE : THE POWER SUPPLY TO ESP8266 IS 3V , SO THE CONNECTION FROM ARDUINO HAS TO BE GIVEN TO THE 3V POWER SUPPLY FROM THE ARDUINO.

To get the logistics usage of the gas, the billing person connects the ESP8266 and acquire the data from the Arduino Leonardo , i.e. the ON time of the pump . He then connects this module to a IP address to send the logistics to the data server . The pump is then reset using an other IP address .

**PRODUCT UPGRADATION/GOALS:**

* Since the Digital literacy and Internet availability in rural areas are less, we shall replace the feature of retrieving the usage logistics of Gau Gas consumed by the customer through a SMS rather over Internet.
* This does not require any internet facility to send the messages, all we need is an active connection.
* These usage logs are encoded and then transmitted to the Database for security purposes.
* A graphical dashboard at the provider’s side provided for easy maintenance of systems.

CLIENT PROVIDER

Gas flow Gas flow

SMS Gateway

Dash Board

Database

GSM Module

Stove Appliance

PUMP

GAS

DIGESTER

Arduino

Mounted On usage text

log file

**OUR APPROACH:**

* Presently they are using duration of pump runs to measure the volume of the gas transfer from digester to the store since the flow rate of pump is constant.
* So whenever the pump is switched on a system uptime should be stored in the memory, whenever the pump is switched of then the system downtime should be stored.
* This usage logs is transmitted to the database using a GSM module.
* Usage logs are transmitted per day or per hour as per provider requirements.
* If the usage logs are received from same number then it should be added to the previous logs of that number.
* We use a SMS gateway for transmission purpose and the usage logs are encoded before transmission.
* Dashboard/statics display are used to represent the usage logs.
* Based on these readings the billing company can prepare the required bill for the customer.

Functional Requirements:

<Req\_1>

Switch Off and On the pump

<Req\_2>

Logging the UP and Down Time of Pump

<Req\_3>

Transmit the usage statistics to the remote DB through SMS.

<Req\_4>

Text encryption and decryption.

<Req\_5>

Dashboard/Statistics Display

<Req\_6>

SMS Alert to the Provider and Client.

<Req\_7>

Generate automated bill on usage logs.

**NON – FUNCTIONAL REQUIREMENTS:**

### Performance:

The system must be interactive and the delays involved must be less .So in every action-response of the system, there are no immediate delays. In case of opening windows forms, of popping error messages and saving the settings or sessions there is delay much below 2 seconds, In case of opening databases, sorting questions and evaluation there are no delays and the operation is performed in less than 2 seconds for opening ,sorting, computing, posting > 95% of the files. Also when connecting to the server the delay is based editing on the distance of the 2 systems and the configuration between them so there is high probability that there will be or not a successful connection in less than 20 seconds for sake of good communication.

### Safety:

Information transmission should be securely transmitted to server without any changes in information.

**Reliability:**

As the system provide the right tools for discussion, problem solving it must be made sure that the system is reliable in its operations and for securing the sensitive details.

**Availability:**

If the SMS service gets disrupted while sending information to the server, the information can be send again for verification.