### GAU GAS METERING AND BILLING AUTOMATION

### Software Requirements Specification

### <<Version 1.0>>

### Participants

**Vinay Kiran M**

**Perumal S**

**Tilak K**

**Vinay Bhat**

Contents

**DESCRIPTION**

This recommended practice describes recommended approaches for the specification of software requirements. It is based on a model in which the result of the software requirements specification process is an unambiguous and complete specification document. It also provides us the details about functional and non-functional requirements .

Sustain Earth uses innovative Biogas technology to provide affordable, clean cooking gas to 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 reliable, easier to maintain. They use Internet and Mobile connectivity to keep track of the performance of their systems.

The flow sensor is used to detect and assure that the constant flow rate of gas is maintained properly .

The main disadvantage of this is that the readings should be manually noted down in each and every house the systems have been installed and based on these readings which the billing is done by the company.

Since these setups are usually present in rural areas it is not possible to use internet for serving this purpose as internet is not that reliable in rural areas as many might not know the usage and working of the internet .

So our main challenge and aim is to minimize the man work done by the company to collect the usage logistics of the gas meter and to implement automation and also to design it in a more cost effective way to measure the gau-gas used by the consumer.

Necessary Requirements:

This practice describes us the required functional and non functional requirements of the model.

The functional requirement defines a function of a system or its component . A function is described as a set of inputs , the behavior and outputs . The plan for implementing these functional requirements are :

**Software Requirements:**

* **System Software :** Windows 7/8/8.1/10 , Ubuntu 14.01
* **Application Software :** Arduino IDE, C Complier
* **SMS GateWay :** Yet to be decided
* **Programming Languages :** AVR C , C

**Hardware Requirements:**

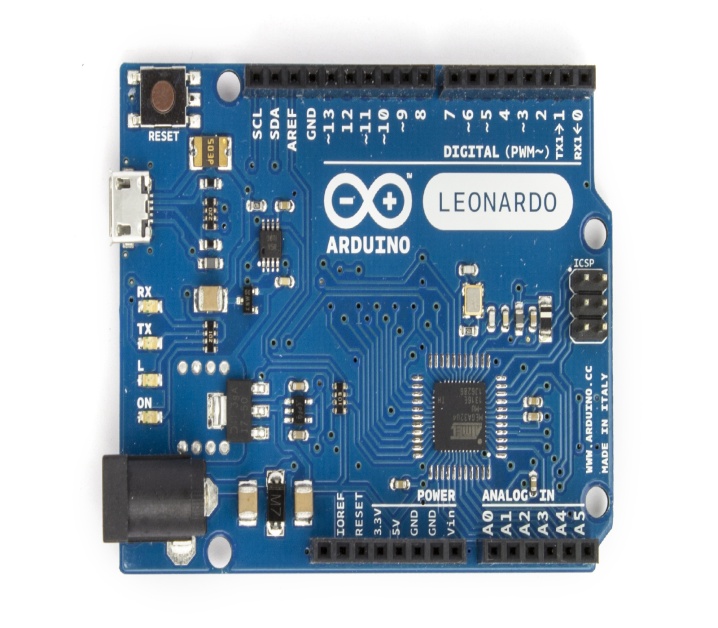
* **Microcontroller :** Arduino Leonardo
* Connecting Wires
* GSM Module
* ESP8266 wifi module(to be eliminated)
* **System Memory :** minimum 512 mb of ram
* **Secondary Memory :** 8 gb of free space in any drive of minimum 40 gb .
* Power Source of 5v to arduino Leonardo Board.

The above mentioned specifications shall provide a facility to the way of approach to the model. However , the functionality and working of the model is stated as below .

**PRODUCT FUNCTIONS:**

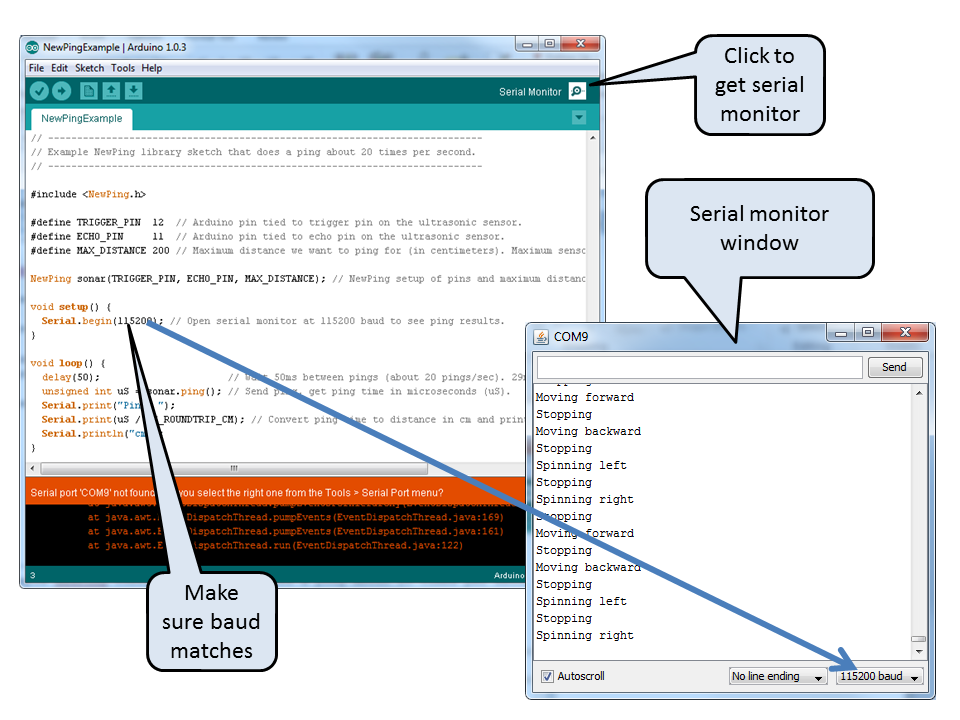
We select the arduino Leonardo board to calculate the total run time of the pump . The Arduino Leonardo is a microcontroller board based on the ATmega32u4 . It has 20 digital input/output pins (of which 7 can be used as PWM outputs and 12 as analog inputs), a 16 MHz crystal oscillator, a micro USB connection, 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 .



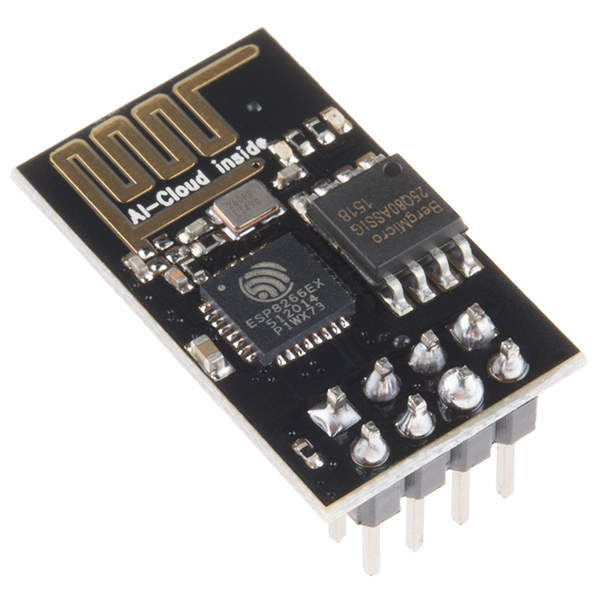
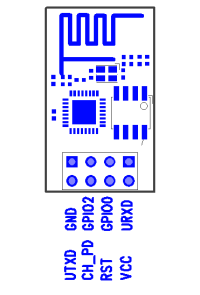
**Constraint of Arduino Leonardo:**

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.

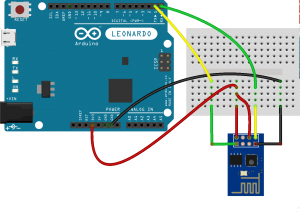
So , the usage logistics of the gas cannot be retrieved when the device gets reset .To , overcome this drawback of leonardo we use the ESP8266 wifi module .

**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 comunication. 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:**

Since the usage of Internet is not known for the people living in the rural areas , we shall replace the feature of retrieving the usage logistics of Gau Gas consumed by the customer through a SMS rather than Internet . The consumer might be able to understand the message that is being transmitted to the data server through a GSM module .

This module does not require any internet facility to send the messages , all we need is an active connection .

**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 message should be sent to an simple application which in turn triggers a counter whenever the pump is switched off then the counter stops and then the data is stored in server if the message is received from same number then the value of the counters should be added or the time between on and off pump run should be saved in microcontroller itself and then for some time interval it should be sent as a message to server.

**Implementation:**

The usage logistics of the gas is collected based on the time that has been used by the customer.

The difference between the time ON and OFF is calculated as the volume of gas consumed by the customer . The GSM on the other hand that resides on the Arduino Board sends the time ON and OFF to the counter (which is basically an application that gives us the details about the usage logistics of the volume of the gas consumed) . This Counter keeps on increasing till the end of the day and the total counter time of the whole day is sent to the data server for billing purpose to the company .

The purpose of not using the internet is solved and the usage logistics is directly sent to the server without any requirement of billing man . The usage logistics should be analyzed by using the data analytics part and a graphical representation of this usage logistics should be plotted in order to make the information more accurate to the billing company .Based on these readings the billing company can prepare the required bill for the customer .

**NON – FUNCTIONAL REQUIREMENTS:**

A **non**-**functional requirement** is a **requirement** that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors .In order to maintain the model these requirements are listed below :

### 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.