

## **TITLE: FIRMWARE DESIGN AND CODE**

### **Design Objective:**

- LCD (Liquid Crystal Display) display to show periodic energy consumption, credit remaining and tariff.
- Interface to Cicada communication module assuming GSM connection.

### **Design specification and justification**

In the design of the GSM based prepaid energy meter, several meter designs were investigated, in order to have an historical experience on energy meter design in terms of design flaw, topology, advantage of one design over the other, NEMSA certification design guide/rule.

#### **1. LCD DISPLAY.**

LCD was incorporated into the firmware code, to enable easy physical communication between users and meter, information can be sent from micro-controller to LCD either using the full 8bit data line or in nibble using 4bit, in this design, data is transmitted from the micro-controller to the LCD in nibbles, as this helps to save the micro-controller 4 extra pins, also the speed difference between 8bit data transmission and 4bit data transmission is in micro-seconds, which is negligible for application such as information display. The micro-controller communicates with the 16x2 LCD screen using M2M (Micro-controller to Micro-controller) I2C (Inter Integrated Circuit) serial communication protocol. This protocol enables communication with several M2M communication, where the micro-controller is the master and the other micro-controller which in our case LCD is the slave. Only the address line and data line (D4, D5, D6, D7) for LCD is needed to tell the address of the slave micro-controller for a particular data sent by the master. Arduino LCD library was used to send information to the 16x2 LCD screen.

#### **2. INTERFACE TO CICADA COMMUNICATION MODULE (GSM).**

GSM internet communication firmware code was written on the Arduino-uno IDE (Integrated Development Environment), featuring the cicada GSM communication module, the internet communication protocol between the GSM and web application is MQTT(Message Queuing Telemetry Transport), this

communication protocol was used due to its speed and reliability in sending and receiving light weight messages with no need for header handshakes. Several library was used to enable easy MQTT communication, library such as : Arduinohttp client, TinyGSm library, pubsubclient library, things board. The serial communication protocol between the Cicada GSM module and the micro-controller is USART(Universal Synchronous Asynchronous Receive Transmit), this protocol enables fast communication between the GSM module and the micro-controller, only two pin is required for communication between the micro-controller and GSM module (Rx & Tx), the receive pin of the GSM module is connected to the transmit pin of the micro-controller hardware awhile the receive pin of the Cicada GSM module is connected to the transmit pin of the micro-controller. Hardware serial was used to communicate between the micro-controller and GSM module to increase reliability and library conflict as opposed to using software serial.

### **General firmware design specification and justification**

In the development of the GSM based prepaid energy meter, several meter design were put into consideration, in view of this, most part of the code had to be written with as little library as possible, so as to reduce library conflict, increase operation speed, and measurement accuracy, the firmware code was properly structured to give attention to the most important task at a particular time, the GSM communication part is structured in such a way, the module stays connected, and whenever there is top-up, it fetches the top-up and transfers it to the meter, if there is top-up available, it takes about 2 to 3sec to fetch the information and transfer it to the meter, while the energy consumed for a whole day and remaining credit is logged on Things-board web-server.

The AC voltage and current is measured using an AFE(Analog Front End) chip with a high measuring accuracy and error as little as 0.1%, the AFE chip used is ATM90e26.

The credit is deducted based on energy consumed per-time, also the remaining credit is registered to the non-volatile memory of the AT24C32 which is an external (EEPROM) Electrically Erasable and Programmable Memory on the RTC module, the information is saved every 60mins due to EEPROM erase and write time, which is limited to 1,000,000 times per chip life time, with this, we write to the memory of the EEPROM 24times per-day and 8,760 times per year, it will take over 100years for the memory to be full, this chip also has multiple location where information can be saved into.

STS token/keypad was incorporated into the firmware code to increase flexibility and permit users to do a manual top-up in event of network failure. Over voltage and under voltage protection was implemented in the firmware code, with the high voltage threshold being 265V maximum and low voltage being 170V for a 220V power specification, developers can change this limit as desired based on the voltage specification in developers' country.

The code was written in Arduino IDE due to its open-source platform, and ease of use to many people, also STM32 CubeMX was installed as an external patch to help in the upload of the code to the micro-controller.

Finally, RS232 communication port was incorporated to enable easy debugging via a serial terminal application and other uses necessary to a developer/OEM (original electronics manufacturer).