International Journal of Research in Engineering, Science and Management
Volume 4, Issue 1, January 2021
https://www.ijresm.com | ISSN (Online): 2581-5792

Embedded Systems – A look into Optimized Energy Metering Devices

Krupa Khapper*
Department of Electronics and Telecommunication Engineering, Sinhgad Institutes, Pune, India

***Abstract*: Embedded Systems are the future of our technology world. At the core of every significant technological piece lie elements of embedded systems. These range from self-controlled cars, intelligent building designs, automated factory processes, aeronautic gears to smart grid energy systems. As the name implies, Embedded Systems (ES) are systems working within other systems to make them more efficient and effective. This paper elucidates recent technological advancements in this field of Engineering and identifies its relevance to nearly every other area of 21st century technology. To provide a clear understanding of the rising level of relevance of this emerging field of Engineering, this paper further narrows down on the modern energy metering designs which now incorporates embedded systems dynamics to improve its functionalities thereby painting a vivid picture of the positive effects delivered by ES Engineering.**

***Keywords*: Embedded systems, energy meter, microcontroller, programming.**

1. Introduction

Technology has evolved from the development of rigid mechanical and electrical solutions to consumer problems. Engineers have recently found interests in developing fluid solutions which can work with other adaptive features in achieving more refined results. Modern cars have transitioned from being purely electromechanical machines to masterpieces housing numerous microcontroller devices computing millions of lines of codes. Research shows that the typical new-model vehicle comes with over 100 million lines of code [1] which run within its various embedded controllers, all performing specific functions that increase its efficiency. This same technology is applied in the aviation, medical, agricultural as well as the oil and gas industries, to mention only a few.

It is important to note that the micro computing controllers which work hand in hand with the other mechanical, electrical and electronic components make up what is referred to as the embedded system. Embedded systems have over the years been defined to be computer systems having dedicated functions within a larger mechanical or electrical system, often with real-time computing constraints [2], [3]. They are embedded as part of a complete device often including hardware and mechanical parts. Embedded systems control many devices in common use today [4]. About 98% of all microprocessors being manufactured are used in embedded systems [5]. Being a special purpose computer, embedded systems are usually completely encapsulated by the device they control and unlike general-purpose computers; they perform pre-defined tasks, usually with very specific requirements. Also, since the system is dedicated to a specific task, design engineers can optimize it, reducing the size and cost of the product, as they are often mass-produced. By so doing, the cost savings may be multiplied by millions of items [6].

As this paper aims to show the improvements this field of Engineering has brought to the technology domain, it narrows down on the Power Systems Digital Energy Metering subject. Energy is a product of work. It is generally defined as the amount of power used/expended over a time range. In Electrical Engineering, Electrical Energy is defined as an electric charge that lets work be accomplished [7]. It is expedient to understand that electrical energy in this case is dependent on a series of analogue variables, primarily current and voltage. These two variables are the active elements used in the derivation of the energy usage per time. Generally, electricity meters operate by continuously measuring the instantaneous voltage (volts), current (amperes) and finding the product of these two to give instantaneous electrical power (watts) which is then integrated against time to give energy used (Joules, Kilowatt-hours etc.). These energy meters are mainly positioned to enable easy quantification of the energy usage per time. In other applications, it helps in the systematic pricing of energy consumed by individual consumer.

The concluding part of this paper explains the methodology adopted in the design and simulation of the modern-day energy metering system founded on embedded systems dynamics. The paper will also juxtapose its benefits, advantages and improvement areas against the previously employed analogue metering system, thereby validating the positive impact offered by intelligent application of Embedded Systems Engineering in Power Systems

2. Similar Works

In the year 2009, Sukriti Jalali of TATA Consultancy Services released a white Paper on „The Trends and Implications in Embedded systems Development“ the paper focused on a solid introduction to embedded systems, including their main components and application areas. It also provided an overview of the emerging trends and the related implications

*Corresponding author: krupa.math@gmail.com

in the design and development of these systems. Sukriti is an Engineer with over 15 years of industry experience in the design and development of real-time embedded systems as applied to a variety of domains including industrial automation, automotive electronics, transportation and process control on [8].

Ogungbenro, K. C. Okafor published a peer reviewed paper on „Digital Metering System: A Better Alternative for Electromechanical Energy Meter in Nigeria". The paper expatiates on the energy metering technology which were being used in some parts of Nigeria, although found to be highly unreliable, thereby requiring substantial labor and time to read, calculate and distribute bills. They highlighted the need to digitize the existing Power Holding Company of Nigeria (PHCN) analogue meter and the increasing demand for smart energy compatible meter. The paper explicitly showed the design process for a low-cost digital meter.

Shifting from the analogue formalities, the device incorporates voltage and current sensors and signal conditioning, all built from discrete components, PIC and liquid crystal display unit. The PIC incorporates both a Peripheral Interface Controller and a ten-bit analogue to digital converter. As the PIC is an embedded program driven device, it is programmed in C language. The design displays power consumption per time as well as expected bill to be paid [9].

3. Embedded Systems Overview

Embedded systems call for real-time operation, reliability, maintenance and cost-effectiveness, which therefore places heavy demands on software (user interfaces, data processing, machine control) and hardware (I/O, Asics, DSP, FPGA). This explains why Embedded Systems is broadly divided into two major parts, as follows:

1. Embedded Software and
2. Embedded Hardware

The FPGA design for system on chip and PCB Design etc., are two major segments of Embedded Hardware. Embedded Software involves Software development which includes mobile application development and Embedded Operating Systems. The Operating System (OS) is one of the most important middleware components that abstracts the underlying hardware and presents a simplified interface to the software application. In embedded systems such as smart-phones, automotive, and avionics, the OS also presents a simplified interface to the multitude of sensors and actuators that these systems interact with. Such systems are highly resource-constrained; therefore, the OS must be efficient in processor and memory usage. Additionally, the ES OS must also support real-time scheduling, to provide service guarantees on systems" timing constraints.

A. Core Components

The major building blocks of an embedded system are listed below:

1. Microcontrollers/digital signal processors (DSP)
2. Integrated chips Real time operating system (RTOS) - including board support package and device drivers.
3. Industry-specific protocols and interfaces
4. Printed circuit board assembly [8]

B. Application Areas

Embedded systems are deployed in various applications and span all aspects of modern life. Fig. 1 below details the main application areas of embedded systems.

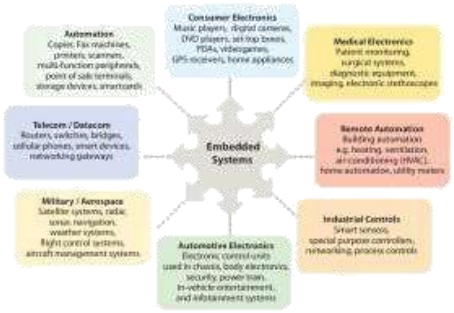

Fig. 1. Major application areas of embedded systems [8]

C. The African Landscape on Embedded System Engineering

Statistics show that the African continent is yet to fully tap into this emerging field of Engineering. Currently, no African Higher Institution undertakes Embedded Systems Engineering as a full scale undergraduate or postgraduate course. At best, it is offered as a module within the main course, carrying very few credits and only covering a handful of modules necessary in the field of study. Although technical schools exist which offer few months of training as well as certification courses in this field, it is penitent to note that there is a complete absence of a degree awarding institution in Embedded Systems Engineering currently in Africa (2015). Consequently, the number of Embedded Systems professionals in the continent are in hundreds or at best, a few thousands and major embedded systems projects are usually outsourced to more developed companies in Europe.

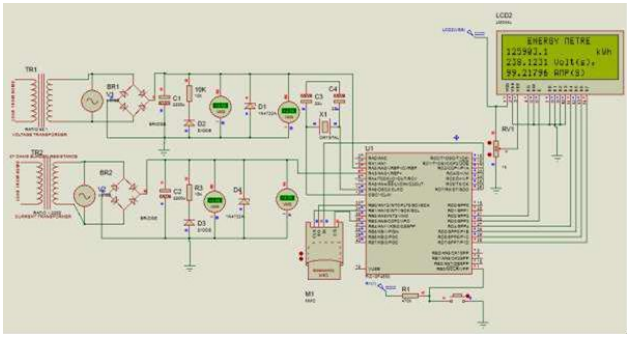

Fig. 2. Schematic of the Digitalized Energy Meter

4. Case Study: Design of Modern Digital Metering System Via Prime Application of Embedded Systems

One key aspect of Embedded Systems Engineering is its link to Digital signal Processing. It is important to note that computers work with digital signals and the processing and generation of results in digital form empowers both user and developer to assess numerically the data being processed. This explains why the digitalization of the metering system results

in the development of a much better and user-friendly device Primarily, the high-resolution sigma-delta Analogue-to-Digital conversion capability of the PIC Microcontroller is used to digitize voltage and current sensed. These values are then appropriately manipulated to get their decimal equivalents. Computing the product of the decimal voltage and current gives the instantaneous power in watts and its integration over time gives energy used, which is usually measured in kilowatt hours (kWh).

The sections below shed more light on the various stages involved in the metering process.

A. Voltage Sensing & Transformation Stage

Many electrical signals around us are analogue in nature. That means a quantity varies directly with some other quantity. The first quantity is mostly voltage while that second quantity could be temperature, pressure, light, force or acceleration.

In the design of the embedded systems digitalized meter, the voltage sensing is done using a voltage step down transformer designed to have a primary to secondary winding ratio of 55:1. The voltage from the power distribution box of the home is connected to the primary winding of this transformer.

$$Vp/Vs = Np/Ns \quad (1)$$

$$V_{PEAK} = \sqrt{2} * V_{RMS} \quad (2)$$

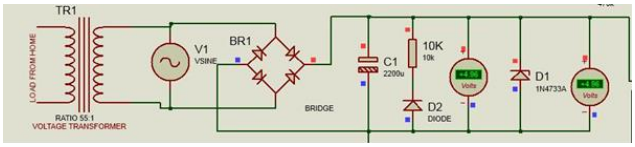

Fig. 3. Schematic of the Voltage Transformational System

The Energy meter is designed to measure a maximum of 240V, 100A AC signal. This voltage is then transformed into an equivalent between the ranges of 0 to 5V DC. With the transformation system in place, other inputs between 0 and 240V AC will also be automatically transformed to their 0-5V DC equivalent. This is because the maximum voltage level the analogue pin of the PIC microcontroller can measure is 5V DC.

Using 240V input as an illustration, from equation 1 above, we can deduce that the transformer will step down an input voltage (Vp) of 240V AC into 4.35V R.M.S at its secondary terminal and also, from Equation 2, we can calculate the peak output voltage of the transformer to be 6.2Vpeak. This peak secondary terminal output voltage passed through the rectifier circuit will drop 1.2Volts, thereby delivering a total of about 5V DC to the analogue pin of the MCU.

B. Current Sensing & Transformation Stage

For current sensing and transformation, a toroidal current transformer is used. The current transformer (C.T) is an "instrument transformer" that is designed to produce an alternating current in its secondary winding which is proportional to the current being measured in its primary.

$$Turn\ Ratio = n = \frac{N_P}{N_S} = \frac{I_S}{I_P} \quad (3)$$

Therefore,

$$Secondary\ Current = I_S = I_P\left(\frac{N_P}{N_S}\right) \quad (4)$$

The maximum expected input current flow that can be measured by the energy meters is pegged at 100 Amps, therefore, the toroidal current transformer is designed with 1 turning in its primary coil, and 2000 in its secondary coil. Substituting these winding parameters into equation 4 above, at a maximum primary current of 100A in the primary terminal, 50mA will be produced in the secondary. The current transformer is also designed to have an inbuilt burden resistor of 87.6 ohms, hence, the 5A produced, will generate 4.384V RMS which translate 6.2V peak by applying (2) above. As in the case of the voltage sensor, the rectifier circuit drops 1.2V out of the 6.2Vpeak, leaving the output at about 5V DC. This 5V DC is then supplied to the AN3 pin of the PIC18F4550 Microcontroller.

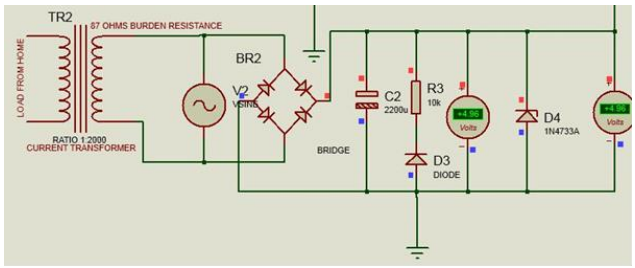

Fig. 4. Schematic of the Current Sensing and Transformation Circuit

C. Rectification Stage

The bridge rectifier consists of 4 diodes. It rectifies the alternating signal into a direct signal. The 2200uF electrolytic capacitor helps to stabilize the output signals by charging and discharging at up and down times respectively. The 1N4733A is a 5.1 V Zener diode, which is added in parallel to these analogue input pins to protect PIC from over voltages. Note that the bridge rectifier always drops an average of 1.2 volts from the signal passed through it.

D. PIC High Resolution Analogue to Digital Conversion & Corresponding DC Signal Interpolation Stage

To have a full understanding of the ADC conversion, the subsections below give detailed explanations.

1) Introduction to Microcontrollers

A microcontroller (MCU) is a small computer on a single chip. Synonymous to its name is its function. Microcontrollers are used to execute relatively simple tasks within a system. A higher grade of these are the microprocessors which are far more sophisticated and can execute numerous operations, simultaneously (such as you have in your computer). A microcontroller is also called an embedded controller because the microcontroller and its support circuits are often built into, or embedded in, the devices they control [10].

2) Digital Nature of MCU and Resolution

MCU are digital in nature. They can only differentiate between HIGH or LOW level on input pins. For example, if an input is more than 2.5V it will be read as 1 and if it is below 2.5 then it will be read as 0 (in case of 5v systems). So, we cannot measure voltage directly from MCUs. To solve this problem most modern MCUs have an Analogue to Digital Converter (ADC) unit. This helps in converting a voltage value to a number so that it can be processed by a digital system like MCU [10].

MCU ADCs come in varied specifications mainly differentiated by their resolutions (8bit, 10bit, 12bit etc). Because the MCU analogue pins can only measure a range of 0 to 5V input voltage, the level of precision to which this voltage can be measured is highly dependent on the resolution of the ADC. For example, an 8-bit ADC will successfully break the 0 to 5V range to 256 different levels, meaning, it can measure accurately, a 19mV change in voltage level conveniently. Likewise, a 10-bit ADC will divide the measurement range to 1024 various levels, meaning a 5/1024 = 4.8mV change can be successfully detected.

$$RESOLUTION = \frac{(v_{ref+} - v_{ref-})}{(Maximum\ ADC\ Value - 1)} \tag{5}$$

Vref+ - Vref-=5 by default for MCU, except changed.

$$RESOLUTION = \frac{5}{Maximum\ ADC\ Value - 1} \tag{6}$$

In a 10-bit ADC, for every 4.887mV change in input voltage level, the ADC value changes by 1. To understand the ADC conversion mechanism, it is imperative to learn the meanings of core ADC terms as explained below.

1. *ADC Reference Voltage:* The reference voltage specifies the minimum and maximum voltage range of analogue input.
2. *ADC Acquisition Time:* The time taken to fully charge the internal holding capacitor when a specific channel is selected.
3. *ADC Clock:* It is the time required to generate 1 bit of conversion.

3) Digital Conversion and Interpolation Proper

The ADC used for the HEMS Energy meter is a 10bit ADC. The read ADC value is converted to its 5v equivalent by the equation below.

$$0 - 5\ volts\ equivalent = \frac{Read\ ADC\ Value}{204.6} \tag{7}$$

This is because the maximum possible ADC reading for a 10bit ADC is 1023, and 1023/204.6 = 5, therefore, 204.6 is the divider value necessary to get the 0 to 5-volt equivalent of the ADC reading (for a 10-bit ADC).

The firmware codes in the MCU help carry out the digital interpolation, considering the prior understanding that a 5V input equates 100A input. In through simple arithmetic calculations, the 0 to 5V equivalent is then transformed to its initial value i.e., its initial value before being passed through the transformational circuit. The formula below explains how this is achieved:

Voltage:

$$Home\ load\ voltage\ drop = \frac{(0\ to\ 5\ Volt\ Equivalent) * 240}{5} \tag{8}$$

Current:

$$Sum\ total\ current\ flow\ in\ the\ home = \frac{(0\ to\ 5\ Volt\ Equivalent) * 100}{5} \tag{9}$$

E. Power & Energy Conversion

$$Electrical\ Power = Voltage * Current \tag{10}$$

$$P = I * V\ watts, where\ wats = Joules/Second \tag{11}$$

Electrical Energy E is calculated either in kWh or in Joules.

$$E_{kWh} = \frac{I_{amps} * V_{volts} * t_{hours}}{1000} \tag{12}$$

The transformed voltage, current and microcontroller delay timer are used to calculate the instantaneous energy usage at every moment.

F. LCD Interfacing Stage

The Liquid Crystal Display (LCD) LM044L is used to display the meter reading. This LCD has 4 visible display rows.

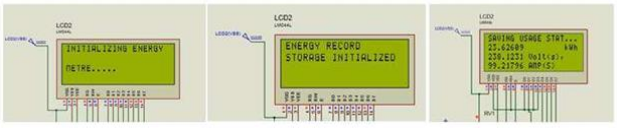

Fig. 5. Image of the LCD Module in Operation

G. Storage Stage

The readings of the energy meter are saved in a series of .txt files created in the Multimedia Card (MMC) Storage incorporated into the Energy Meter module; This storage is done hourly, daily, weekly, monthly and yearly. The data saved can be accessed by an interfacing desktop application.

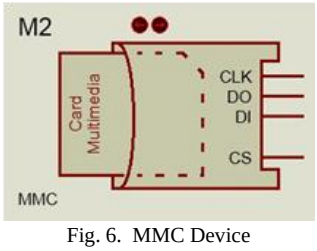

Fig. 6. MMC Device

H. Firmware Code

Below is the full firmware code for the PIC18F4550 Microcontroller. The compiler used is the MikroC Pro

Compiler and language deployed is the C programming language.

Code 1: Firmware Code for the Embedded Microcontroller Operation

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <float.h>
#define INT_RANGE 1000
#define DEC_RANGE 10


// LCD module connections
sbit LCD_RS at RD2_bit;
sbit LCD_EN at RD3_bit;
sbit LCD_D4 at RD4_bit;
sbit LCD_D5 at RD5_bit;
sbit LCD_D6 at RD6_bit;
sbit LCD_D7 at RD7_bit;
sbit LCD_RS_Direction at TRISD2_bit;
sbit LCD_EN_Direction at TRISD3_bit;
sbit LCD_D4_Direction at TRISD4_bit;
sbit LCD_D5_Direction at TRISD5_bit;
sbit LCD_D6_Direction at TRISD6_bit;
sbit LCD_D7_Direction at TRISD7_bit;
// End LCD module connections

//MMC Module Connection
sbit Mmc_Chip_Select at RB3_bit;
sbit Mmc_Chip_Select_Direction at TRISB3_bit;

//Declaration of Variables
unsigned int v,i,verificationformat, filehandler, k;
unsigned long time=1;
unsigned int dayCounter=1,weekCounter=1, monthCounter=1,
yearCounter=1;
char *voltageTXT[11], *currentTXT[11],
*energyKWHTXT[11], *energyJoulesTXT[11];
char *hourCounterTXT[11], *dayCounterTXT[11],
*weekCounterTXT[11];
char *monthCounterTXT[11], *yearCounterTXT[11];
float Joulespower, Joulespowerbefore=0, energyKWH,
energyJOULES;
float voltage, current,KWHpower, KWHpowerbefore=0;
 char txt1[] = "INITIALIZING ENERGY METRE";
 unsigned char readbuff[256], writebuff[256];
 char newline = '\n', carriagereturn='\r';
//End of Variable Declaration

//Function for reading voltage values through the ADC Analogue
  terminal
 void get_Voltage(){
 v = ADC_Read(2); //Get ADC value from the voltage input at pin AN2
 voltage = (float) (v/204.6);
 voltage = (voltage*240)/5; //Because 5V == 240Volts
 }
//End of function

/*Function for reading voltage values equivalent to the current flow
through the ADC Analogue terminal */
 void get_Current(){
 i = ADC_Read(3);
 current = (float) ((i/204.6) * 20); //Because 5V =100A
 }
//End of function

//Function for computing power
 void compute_Power(){
 Joulespower = voltage*current; //Watts
 Joulespower = Joulespowerbefore+Joulespower;
 Joulespowerbefore=Joulespower;

 KWHpower =voltage*current; //Watts
 KWHpower = KWHpowerbefore+KWHpower;
 KWHpowerbefore=KWHpower;

 }
//End of function


//Function for computing Energy Expended in kWh
 void get_energyinKWH(float sumpower, long time){
 time = time/3600; //3600 seconds makes an hour
 //sumpower = voltage * current
 /*Energy Usage in watt-hour = Voltage*current*time(in hours)
 Divided By 1000 to make it in to make it in kilowatts */
 energyKWH = (float) (sumpower*time)/1000;
 }
//End of function

//Function for computing Energy Expended in kWh
 void get_EnergyinJOULES(float sumpower){
 energyJOULES = (float)sumpower; //Cummulative watt usage
   persecond
 }
//End of function

//Write to Storage
 void store_energy_Hourly(){
 LCD_Out(1,1," ");
 LCD_Out(1,1,"SAVING USAGE STAT...");
 Mmc_Fat_Assign("UsageH.TXT",0xA0);
 Mmc_Fat_Write(writebuff, 0); //Where file writing begins from
 Mmc_Fat_Append(); //Moves pointer to end of file
 Mmc_Fat_Write("Hour",4);
 Mmc_Fat_Write(":",1);
 Mmc_Fat_Write(energyKWHTXT,11);
 Mmc_Fat_Write(" kWh, ",6);
 LCD_Out(1,1," ");
  LCD_Out(1,5, "ENERGY METRE");
  Mmc_Fat_Append();
  //Mmc_Fat_Write(carriagereturn,1);
  //Mmc_Fat_Write(newline,1);
  //Mmc_Fat_Append();


}
 void store_energy_Daily(){
IntToStr(dayCounter, dayCounterTXT);
Mmc_Fat_Assign("UsageD.TXT",0xA0);
Mmc_Fat_Write(writebuff, 0);
Mmc_Fat_Append(); //Moves pointer to end of file
Mmc_Fat_Write("Day ",4);
Mmc_Fat_Write(dayCounterTXT,6);
Mmc_Fat_Write(":",2);
Mmc_Fat_Write(energyKWHTXT,11);
Mmc_Fat_Write(" kWh, ",6);
dayCounter = dayCounter+1;
}
 void store_energy_Weekly(){
IntToStr(weekCounter, weekCounterTXT);
Mmc_Fat_Assign("UsageW.TXT",0xA0);
Mmc_Fat_Write(writebuff, 0);
Mmc_Fat_Append(); //Moves pointer to end of file
Mmc_Fat_Write("Week ",5);
Mmc_Fat_Write(weekCounterTXT,6);
Mmc_Fat_Write(":",2);
Mmc_Fat_Write(energyKWHTXT,11);
Mmc_Fat_Write(" kWh, ",6);
weekCounter = weekCounter+1;
}
 void store_energy_Monthly(){
IntToStr(monthCounter, monthCounterTXT);
Mmc_Fat_Assign("UsageM.TXT",0xA0);
Mmc_Fat_Write(writebuff, 0);
Mmc_Fat_Append(); //Moves pointer to end of file
```

```
Mmc_Fat_Write("Month ",5);
Mmc_Fat_Write(monthCounterTXT,6);
Mmc_Fat_Write(":",2);
Mmc_Fat_Write(energyKWHTXT,11);
Mmc_Fat_Write(" kWh, ",6);
monthCounter = monthCounter+1;
}
void store_energy_Yearly(){
IntToStr(yearCounter, yearCounterTXT);
Mmc_Fat_Assign("UsageY.TXT",0xA0);
Mmc_Fat_Write(writebuff, 0);
Mmc_Fat_Append(); //Moves pointer to end of file
Mmc_Fat_Write("Year ",5);
Mmc_Fat_Write(yearCounterTXT,6);
Mmc_Fat_Write(":",2);
Mmc_Fat_Write(energyKWHTXT,11);
Mmc_Fat_Write(" kWh, ",6);
yearCounter = yearCounter+1;
}
void main() {
ADCON1 = 0; ///Configure AN pins in Port A as analogue inputs

//Port Configurations
TRISD = 0xFF; // Configure PORTD as input
PORTD = 0xFF; // Configure PORTB as output
TRISC = 0x00; //Configure PORTC as output

TRISA = 0xFF; // Configure PORTA as input
TRISB = 0; // Configure PORTB as output

Lcd_Init(); // Initialize LCD
LCD_Cmd(_LCD_CURSOR_OFF); //Remove LCD Cursor

//Animate System Loading Pattern
LCD_Out(1,1,txt1);
for(k=0; k<5; k++){
LCD_Out_Cp(".");
delay_ms(500);
}
// Initialize SPI1 module and set pointer(s) to SPI1 functions
SPI1_Init_Advanced(_SPI_MASTER_OSC_DIV4,
    _SPI_DATA_SAMPLE_MIDDLE, _SPI_CLK_IDLE_LO OW,
    _SPI_LOW_2_HIGH);
Delay_ms(100);
Mmc_Fat_Init();

  verificationformat = Mmc_Fat_QuickFormat("STORAGE");
  LCD_Cmd(_LCD_CLEAR); //Clear LCD Screen
  Delay_ms(200);
  if (verificationformat==0)
  {
  Lcd_Out(1,1, "ENERGY RECORD");
  Lcd_Out(2,1, "STORAGE INITIALIZED");
  Delay_ms(5000);
  Lcd_Cmd(_LCD_CLEAR);
  Delay_ms(50);

  //Create File for Currently Bought Energy
  Mmc_Fat_Assign("Bought.TXT",0xA0);//File Created to Save
  kWh purchased
  Mmc_Fat_Write("1000", 4); //Signifying that user bought 1000
  kWh

  //Create File to record Yearly Energy Consumption
  Mmc_Fat_Assign("UsageY.TXT",0xA0);

  //Create File to record Monthly Energy Consumption
  Mmc_Fat_Assign("UsageM.TXT",0xA0);

  //Create File to record Weekly Energy Consumption
  Mmc_Fat_Assign("UsageW.TXT",0xA0);

  //Create File to record Daily Energy Consumption
  Mmc_Fat_Assign("UsageD.TXT",0xA0);

  //Create File to record Hourly Energy Consumption
  Mmc_Fat_Assign("UsageH.TXT",0xA0);

  Mmc_Fat_Write(writebuff, 0); //where the writting would start
  from
  }
  LCD_Out(1,5,"ENERGY METRE");
  do {
  FloatToStr(energyJOULES,energyJoulesTXT);
  FloatToStr(energyKWH,energyKWHTXT);
  FloatToStr(current,currentTXT);
  FloatToStr(voltage,voltageTXT);
  LCD_Out(2,1,energyKWHTXT);
  LCD_Out(2, 17," kWh");
  LCD_Out(3,1,voltageTXT);
  LCD_Out_Cp(" Volt(s), ");
  LCD_Out(4,1,currentTXT);
  LCD_Out_Cp(" AMP(S)");

  get_Voltage(); //Get instantaneous voltage reading
  get_Current(); //Get instantanious current reading
  compute_Power(); //compute instantaneous & power
  get_energyinJOULES(Joulespower); //compute cumulative & instant
  energy usage
  get_energyinKWH(KWHpower, time);
  if(time%3600==0){
  store_energy_Hourly(); //Store Energy Usage Statistics to MMC
  every hour
  }

  if(time%86400==0){
  store_energy_Daily(); //Store Energy Usage Statistics to MMC
  every day
  }

  if(time%604800==0){
  store_energy_Weekly(); //Store Energy Usage Statistics to MMC
  every week
  }

if(time%2592000==0){
store_energy_Monthly(); //Store Energy Usage Statistics to MMC every
    month
}

if(time%31536000==0){
store_energy_Yearly(); //Store Energy Usage Statistics to MMC every
    year
}
Delay_ms(850);
/* The computation and storage process is expected to make up for the
remaining 150ms that makes the complete 1 second cycle */
time=time+1;
} while(1);
}
```

End of Code

5. Key Optimization

The digitized energy metering system offers the owner a more user-friendly meter from which concise energy usage reading can be made in numerical values without having to make near estimates as is common with analogue systems used around the world. The system allows the user to see current load status as well as the cumulative energy usage reading.

The Multimedia Memory Card compatibility of the MCU also makes it possible to keep instantaneous energy usage records.

6. Conclusion

The vivid design example and optimized performance results of the digital energy meter which were based on embedded systems implementations is a close representation of the positive effect embedded systems have on the primary systems they function on. Consequently, this field of Engineering has become of special importance. Oil and Gas industries now employ ES to prevent pipeline vandalization, medicine now makes use of embedded artificial human organs, diagnostic equipment, patient monitoring and surgical systems. A larger percentage of Bank's financial transactions and operations are now via cards with embedded electronic chips. Satellite systems, intelligent buildings, home automation, weather systems, flight control systems, aircrafts management systems and many more fields of technology enjoy varying degrees of ES implementations. It can therefore be stated that Embedded Systems technology is and possesses the potential to be the centre of the 21st century world.

References

[1] Many Cars Have a Hundred Million Lines of Cod, http://www.technologyreview.com/view/508231/many-cars-have-a-hundred-million-lines-of-code/

[2] Michael Barr. "Embedded Systems Glossary". Neutrino Technical Library.

[3] Heath, Steve (2003). Embedded systems design. EDN series for design engineers (2 ed.). Newnes. p. 2. ISBN 978-0-7506-5546-0. An embedded system is a microprocessor based system that is built to control a function or a range of functions.

[4] Michael Barr; Anthony J. Massa (2006). "Introduction". Programming embedded systems: with C and GNU development tools. O'Reilly. pp. 1–2. ISBN 978-0-596-00983-0.

[5] Barr, Michael (1 August 2009). "Real men program in C". Embedded Systems Design. TechInsights (United Business Media). p. 2. Retrieved 2009-12-23.

[6] Embedded Computer Systems, http://www.ece.ncsu.edu/research/cas/ecs, Retreived December 2015

[7] Electrical Energy, http://www.yourdictionary.com/electrical-energy

[8] Sukriti Jalali, Trends and Implications in Embedded Systems Development, White paper, 2009.

[9] M.C. Ndinechi, O.A. Ogungbenro, K.C. Okafor, Digital Metering System: A Better Alternative for Electromechanical Energy Meter in Nigeria, International Journal of Academic Research Vol. 3. No. 5. September, 2011.

[10] Oluwole Oyetoke and Adedayo Adedapo, A Microcontroller Based Embedded System Design for Device Automation and Control in Intelligent Buildings, International Journal of Research, Volume 2, Issue 12, December 2015.