Internship Report

Submitted to:

Research and Development Lab
Capital Power Systems Ltd.
Noida, Uttar Pradesh

A Step Ahead

Submitted by:

Raghav Sharma, B. Tech. II year, Electrical Engineering Department
Indian Institute of Technology, Ropar

Supervisor: Mr. VK Mishra

Start Date for Internship: 15th May, 2017
End Date for Internship: 15th July, 2017

Preface

This report documents the work done during the summer internship at Research and Development (R&D) Lab, Capital Power Systems Ltd, Noida, Uttar Pradesh under the supervision of
Mr. V.K. Mishra. The report shall give an overview of the tasks completed during the period of internship with technical details. Then the results obtained shall be discussed and analyzed.

Report shall also elaborate on the future works which can be persuaded as an advancement of the current work.

For any additional inforation, kindly refer the github repository:
Internship Report (https://github.com/Raghav1806/capitalPowerSystems-Summer-Internship)

I have tried my best to keep this report simple yet technically correct. I hope I succeed in my attempt.

Raghav Sharma

Acknowledgments

Simply put, I could not have completed this internship without the guidance and support I received from the Research and Development Lab. The work culture is really motivating. Everybody is such a friendly and cheerful companion here that stress never comes in way.

I would specially like to thank Mr. V.K. Mishra, Mr. Dhananjay Mishra, and Mr. Akhil Dixit for their unconditional support. Not only did they helped me in learning new concepts, but listening to their discussions have evoked a good interest in the field of electronics.

Raghav Sharma

Abstract

The report presents the three tasks completed during summer internship at Capital Power Systems Ltd. which are listed below:

(1) Meter Testing
(2) Circuit design and fabrication
(3) Designing the required software

All these tasks have been completed successfully and results were according to expectations. The main aim of all these steps was to make sure that the final product should follow all the guidlines stated by various state electricity board with minimal failure rate.

Raghav Sharma

Capital Power Systems Ltd.

(https://www.capitalpowers.com/)

As leaders in manufacturing industry, Capital has set the bar with its unwavering commitment to quality and exceptional client service. Through its extensive inventory of diverse and customizable products, the talented team goes above and beyond to make sure that customer's needs are met, and expectations are exceeded.

History and growth:
Capital started with a modest means but with a vision to be best in the industry, and the company is now three decade old. From electromechanical meters manufacturer, the company has has taken a long and strong leap forward to now a smart meter and solutions oriented company.

Capital has expanded its portfolio of services from just manufacturing to developing smart meters, tailored made end to end AMI solutions, turnkey projects and OEMs for its expanding customer base. The company believes that keeping itself updated with latest trend is the only way forward. So the company ensures that its team has right exposure to understand its customers, suppliers by attending national and international exhibitions.

Products:

(1) Energy Meters
(2) Water Meters
(3) Gas Meters

Services provided:

(1) Meter Installation
(2) Meter Reading
(3) End-to-End Advanced Metering Infrastructure (AMI) system

Smart Meter: An Introduction

A smart meter is an electronic device that records consumption of electric energy in intevals of an hour or less and communicates that information at least daily back to the utility for monitoring and billing. Smart meters enable two-way communication between the meter and the central system. Unlike home energy monitors, smart meters can gather data for remote reporting. Such an Advanced Metering Infrastructure (AMI) differs from traditional Automatic Meter Reading (AMR) in that it enables two way commnications with the meter. Communications from the meter to the network can be done via fixed wired connections or via wireless. In using wireless, one can opt for cellular communications, WiFi, low power long range wireless, ZigBee, etc.

Different categories of smart meters:
(1) Category A: Used at sub-station feeders and distribution transformer centers.
(2) Category B: Used at meter banks and network boundaries (import/export of energy).
(3) Category C: For consumers who draw energy from grid.

Meter Testing

Their are two types of meter testing:
(1) Acceptance Test
(2) Routine Test

Following tests are generally carried for testing Electric Meters:

(1) AC high voltage test
Significance: Checks whether meter works perfectly when abruptly large value of voltage is supplied.

(2) Insulation test
Significance: Checks whether meter works perfectly when large voltage is struck on its surface.

(3) Test on limits of error
Significance: Compares the supplied (actual) value of current and voltage applied to the meter given that various combinations of load (resistive, capacitive and inductive) are applied to it.

(4) Test of meter constant
Significance: Compares the number of LED pulses equal to 1 KWHr with the reference value.

(5) Test of starting condition
Significance: Checks whether meter gives reading for a given minimum amount of current (0.5% of I(basic)) given that various combinations of load (resistive, capacitive and inductive) are applied to it.

(6) Test of no-load condition
Significance: Checks the behaviour of meter when voltage is supplied to it, but no load is applied to it.

(7) Test of repeatability of error
Significance: Checks the error produced in meter readings when various combinations of load (resistive, capacitive and inductive) are applied to it.

(8) Test of power consumption
Significance: Checks the power consumed by meter itself for its proper functioning.

Major terms used:

(1) Meter constant: Expresses relation between energy registered by meter and corresponding pulse count of test output.
(2) Basic current: Value of current in accordance with which the relevant performance of meters is fixed.
(3) %age error: (Energy Registered – True Energy)/True Energy.

Electrical Requirements of the meter:

(1) Power consumption
(2) Influence of supply voltage
(3) Voltage range
(4) Voltage dips and interruption
(5) Start time over current
(6) Influence of self heating
(7) Influence of heating
(8) Insulation
(9) Immunity of earth fault

Electromagnetic Compatibility

(1) Immunity to electromagnetic disturbance
    - Electrostatic discharge
    - Electromagnetic high field
    - Fast transient burst
(2) Radio interference suppression

Metering Parameters

Different types of metering parameters:
(1) Instantaneous parameters
(2) Block profile/load survey parameters
(3) Daily profile
(4) Parameters for accounting/billing
(5) General purpose quantities:
    (a) Name plate details
    (b) Programmable parameters
(6) Event conditions

Circuit Design and Fabrication

Microcontroller used: Freescale (NXP) SC667502 GLHS IN29N (Photon DS)

Main circuit components:

(1) DC supply
Motive: To reduce the input voltage levels and feed it into the main circuit.

(2) Real Time Clock (RTC) battery
Motive: Constant power supply to RTC terminal of microcontroller, whether meter is ON or OFF.

(3) Main battery
Motive: Used as a source for various circuit components like power control, push button, etc.

(4) Neutral missing
Motive: For checking neutral missing tampering condition.

(5) Neutral CT
Motive: To check the neutral current in meter in case of tampering.

(6) Shunt
Motive: To check the phase current in meter in case of tampering.
Note: Total current flowing through meter = Phase current + Neutral Current.

(7) Voltage network
Motive: Used to narrow down voltage and input it to the microcontroller, which is then used to output the voltage on LCD screen.

(8) Power control
Motive: To supply voltage when a switch is pressed.

(9) RS 232 communication
Motive: To transfer data using cable from meter to data accumulator (UART).

(10) Optical + IR communication
Motive: To transfer data using optical sensors from meter to data accumulator.

(11) Magnet sensor
Motive: Used to detect high voltage application for tampering purpose.

(12) EEPROM (Memory)
Motive: Memory to store data.

(13) Push button
Motive: Start the meter and show the values of different parameters when the button is pressed.

(14) Debugger connector (JTAG)
Motive: Used to upload code into the meter

(15) Cover open switch

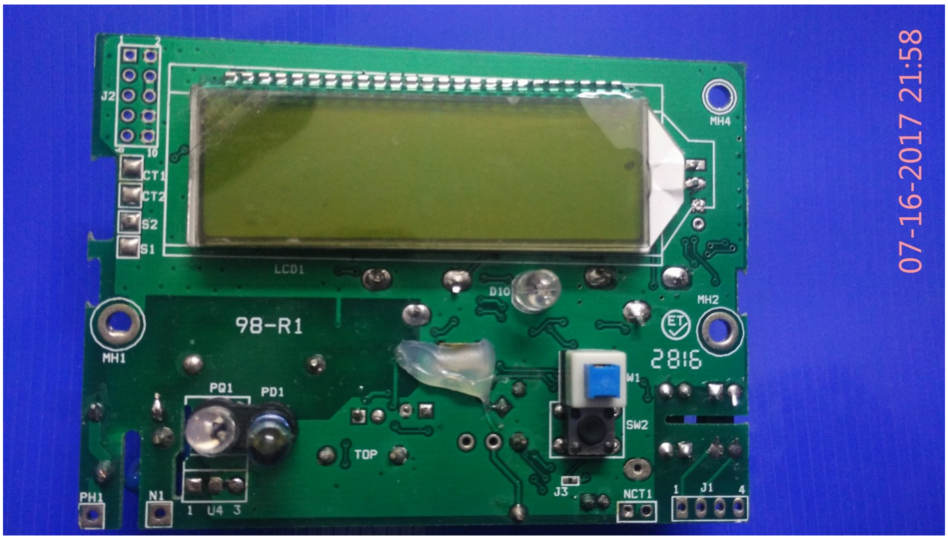
Motive: It checks whether the cover has been opened or not for tampering purpose.

(16) LED

Motive: LED blink rate is directly proportional to user consumption rate.

(17) LCD (25 pin)

Motive: To display various parameters of the meter.



(front view of circuit)

(back view of circuit)

Communication Modes

Their are different possible communication methods through which electronic meters can interact with outside world, generally UART is used in smart electrical meters.

UART: Universal Asynchronous Receiver/Transmitter

- Computer hardware device for asynchronous serial communication.
- Data format and transmission speeds are configurable.
- Acts as an intermediary between parallel and serial interfaces.
- Responsible for both sending and receiving serial data.
- Two UARTs can communicate directly with each other.
- UARTs transmit data asynchronously
- No clock signal is required to synchronize output of bits from transmitting UART to sampling of bits by UART.
- Transmitting UART adds start and stop bits to the data packet being transferred.
- After transmitting UART gets parallel data fro data bus, it adds start bit, parity bit and stop bit, creating a data packet.

Embedded System

Embedded system = Embedded software + Embedded hardware

Common properties of Embedded System:

(1) Low power consumption
(2) Small size
(3) Rugged operating ranges
(4) Low per-unit cost

Peripherals in an Embedded System:
Embedded systems talk with the outside world via peripherals such as:

(1) Serial Communication Interfaces
(2) Synchronous Serial Communication
(3) Universal Serial Bus
(4) Multi Media cards
(5) Networks
(6) Timers
(7) Discrete IO – General Purpose Input/Output
(8) Debugging – JTAG port (Joint Test Action Group)

Embedded Software Architecture:

(1) Simple control loop: A loop calls subroutines, each of which manages a part of the hardware or software.
(2) Interrupt controlled system: Tasks performed by the system are triggered by different kinds of events. Ex: triggers could be generated by a timer in a predefined frequency, or by a serial port controller receiving a byte.
(3) Cooperative multi tasking.
(4) Preemptive multi tasking and multi threading.
(5) MicroKernels and ExoKernels.
(6) Monolithic kernels.

Using EAGLE CAD for designing PCB

Overview:
PCBs are the backbone of every electronic gizmo. They're not flashy like microprocessors, or abundant like resistors, but they're essential to making all components in a circuit connect together just right.

Why EAGLE?

- Cross platform - EAGLE can run on anything: Windows, Mac, even Linux.
- Lightweight
- Free/Low-cost
- Community support

The EAGLE board designer has layers just like an actual PCB, and they overlap too. A palette of colors is used to represent the different layers.

While routing, it’s important to avoid two cases of overlap: copper over vias, and copper over copper. All of these copper traces are basically bare wire. If two signals overlap, they will short out, and neither will do what it’s supposed to do. If traces do cross each other, make sure they do so on opposite side of board.

Checking for errors:

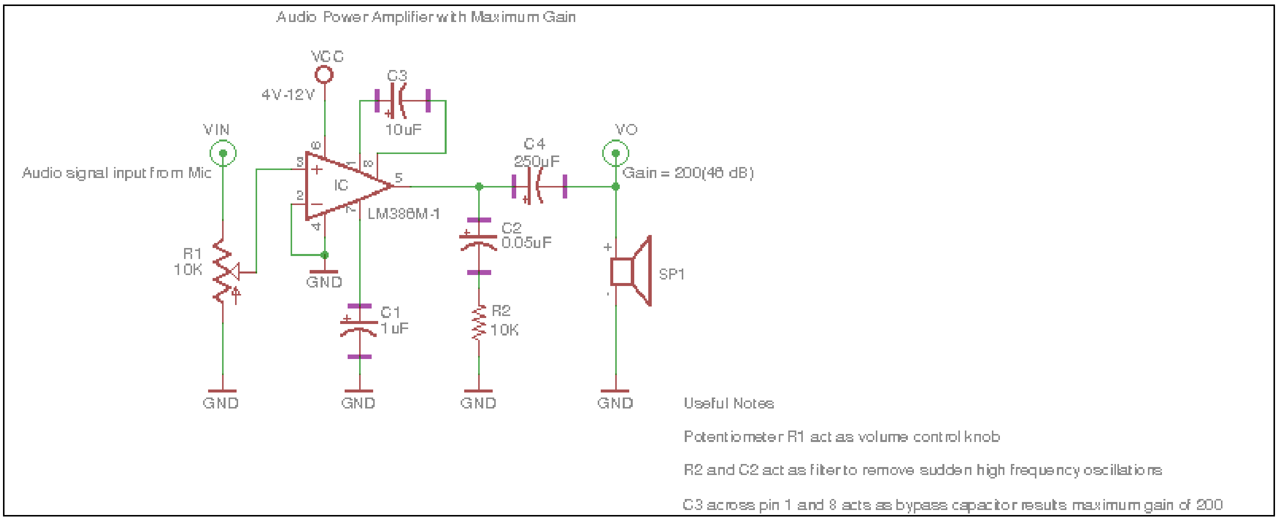
(1) Ratsnest: To make sure you have actually routed all of the nets in your schematic.
(2) Design Rule Check: It is defined as a huge set of design rules which the layout needs to pass

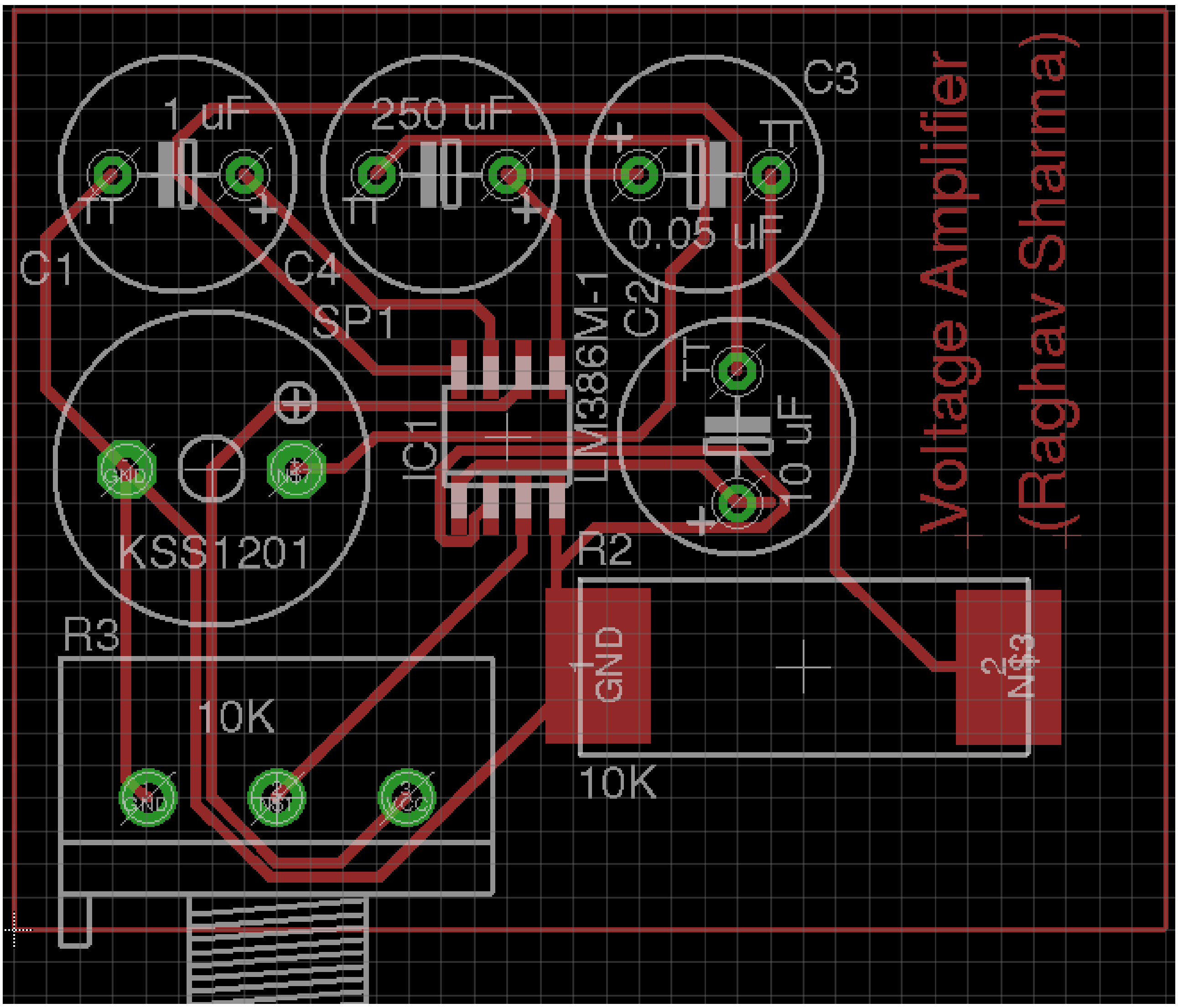
Most common errors in PCB designing:

(1) Clearance: A trace is too close to either another trace or a via.
(2) Overlap: Two different signal traces are overlapping each other, This will create a short if it is not fixed.
(3) Dimension: A trace, pad or via is intersecting with (or too close to) a dimension line. If this is not fixed that part of board will just be cutoff.

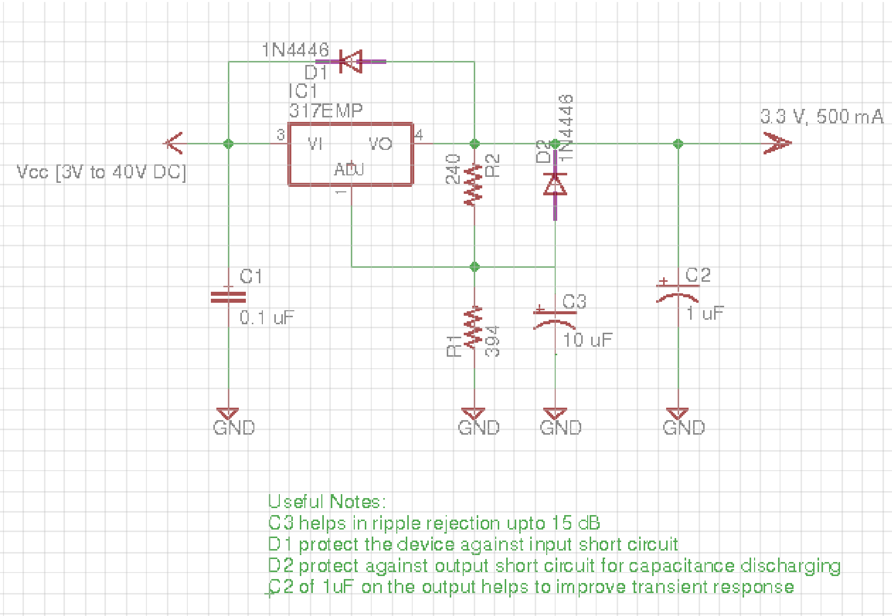
Adding Silkscreen:
Silkscreen can be a critical part of PCB design. We can manually add other information, like labels, logos, and names. a variety of draw tools - wire, text, circle, arc, etc. can be used to draw on the silkscreen layer.

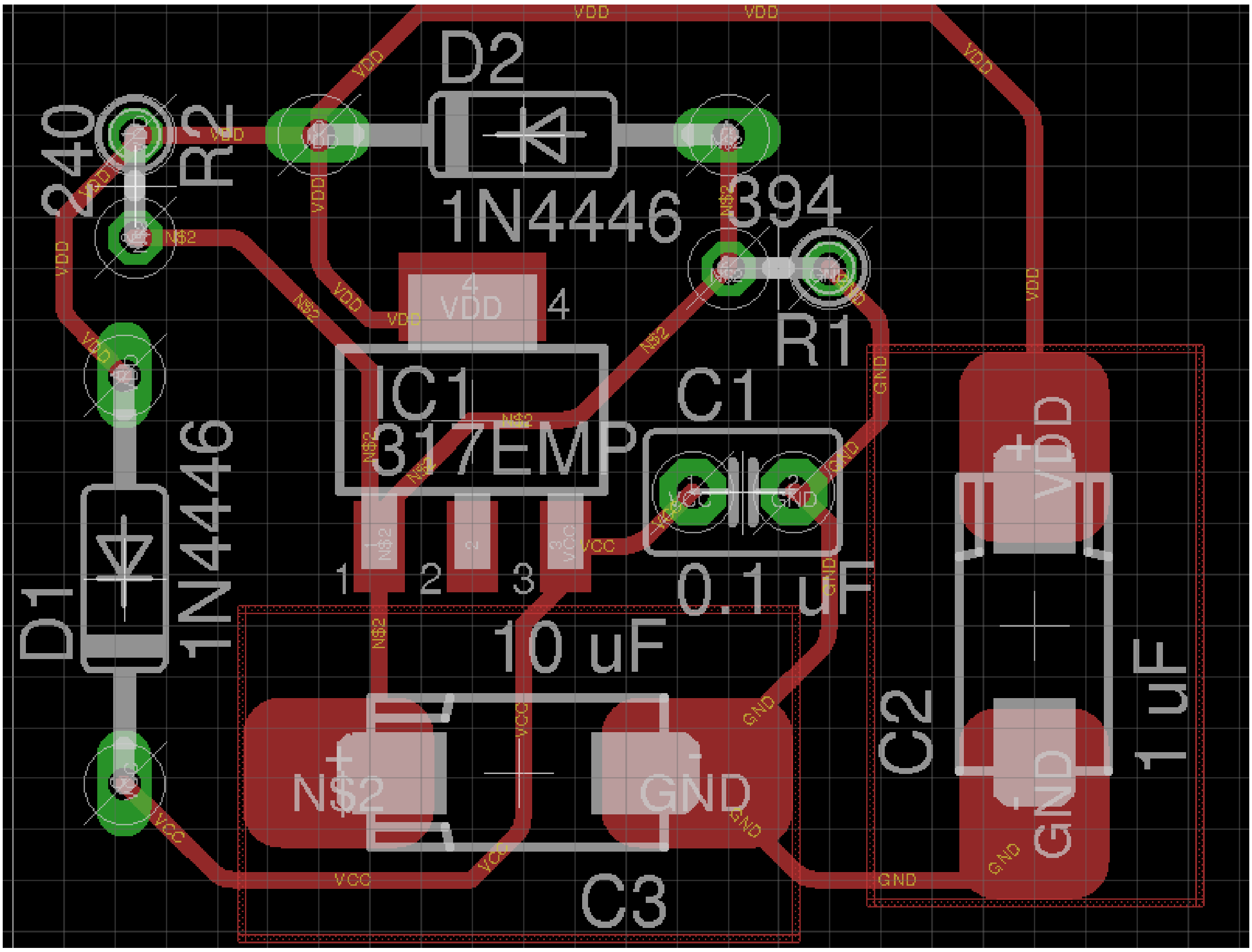
Schematic and Board Layout for voltage amplifier





Schematic and Board layout of voltage stabiliser (3.3 V)





Designing the software for smart meter

General Software flow (pseudo code)
The code is implemented in Embedded C.

```
main(){
        if(AC power == ON){
                RecoverData(){
                        if(NormalMode()){
                                Polling();
                                DisplayInfo();
                                UARTCommandParser();

                                if(TimeChecking() == 1min){
                                        OneMinuteProcess();
                                        if(thirtyMinuteCounter() == true){
                                                getMDData();
                                                recordPeriodLoadData();
                                        }
                                        if(dayChange() == true){
                                                RecordDailyLoadData();
                                                RecordCurrentMonthData();
                                        }
                                        if(monthChange() == true){
                                                recordHistoryMonthData();
                                        }
                                }
                        }
                        else{
                                // tampering case
                                RecordTamperEvents();
                                BackUpData();
                        }
                }
        }
        else if(AC power == OFF){
                // Battery mode
                MidNightEventHandling();
                CoverOpenHandling();
                PushButtonHandling();
                UARTRxWakeUp();
        }
}
```

Polling:

Actively sampling the status of an external device by a client program as a synchronous activity.

(1) The host repeatedly reads the busy bit of the controller until it becomes clear.

(2) When clear, the host writes in the command register and writes a byte into the data out register.

(3) The host sets the command to ready bit.

(4) When the controller senses command ready bit is set, it sets busy bit.

(5) The controller reads the command register and since write bit is set, it performs necessary I/O operations on the device. If the read bit is set to one instead of write bit, data from device is loaded into data-in register, which is further read by the host.

(6) The controller clears the command-ready bit once everything is over, it clears error bit to show successful operation and reset busy bit (0).

Interrupt:

- Signal to the processor emitted by hardware or software indicating an event that needs immediate attention. The processor responds by suspending its current activities, saving its state, and executing a function called interrupt handler (or an interrupt service routine) to deal with the event.

Two major types of interrupts:

(1) Hardware Interrupts: Used by devices to communicate that hey require attention fro the operating system.These interrupts are asynchronous and can occur in middle of instruction execution. The act of initiating a hardware interrupt is referred as an interrupt request (IRQ).

(2) Software Interrupts: Caused either by an exceptional condition in the processor itself, or a special instruction in the instruction set which causes an interrupt when it is executed.

Each interrupt has its own interrupt handler. The number of hardware interrupts is limited by the number of interrupt request (IRQ) lines to the processor, but there may be hundreds of different software interrupts.

Interrupts can be categorized into these different types:

(1) Maskable Interrupt: A hardware interrupt that may be ignored by setting a bit in an interrupt mask register's bit mask.

(2) Non Maskable Interrupt: A hardware interrupt that lacks an associated bit mask, so that it can never be ignored. It is used for highest priority tasks such as timers (watch dog timers).

(3) Inter Processor Interrupt (IPI): A special case of interrupt that is generated by one processor to interrupt another processor in a multi-processor system.

(4) Software Interrupt.

(5) Spurious Interrupt: A hardware interrupt that is unwanted.

Transition between operating modes:

(1) Normal mode to battery mode:

If Vdd < 2.75V, enable battery switchover events occur. The ADCs will be disabled by default. To reduce power consumption, the user code can initiate a transition to sleep mode.

Entering Sleep Mode: To reduce power consumption when VddSWO is connected to VddBAT, user code can initiate a transition to sleep mode.

(2) Sleep mode to battery mode:

Microcontroller may need to wake up from sleep mode to service wake-up events, and can return to sleep mode by setting registers to shut down the the microcontroller core.

(3) Sleep mode to normal mode:

If Vdd > 2.75V, enable battery switchover events occur, VddSWO switches to Vdd. When this switch occurs, the microcontroller code can initiate normal operating mode by identifying VBAT_SWITCH flag.

(4) Battery mode to normal mode:

If Vdd > 2.75V, before microcontroller enters sleep mode, the operating mode switches to normal mode. When this switch occurs, the code execution continues normally. A software reset can be performed to start from the beginning.

Interrupt Service Routine

A microcontroller is designed with a clean and efficient procedure to handle interrupts.

(1) If a peripheral's interrupt enabled, the appropriate signal sets the interrupt flag in one of the p eripheral's registers.
(2) If the processor is allowing interrupts, the presence of a peripheral interrupt flag sets a g eneral flag to the processor.
(3) The processor finishes any instruction it's currently doing.
(4) The processor saves its place by copying the address where it's currently executing instructions (the value of Program Counter register) to the memory stack.
(5) The processor saves its current status by pushing the status register onto the stack.
(6) The interrupt that currently has the highest priority is selected if there are multiple flag set.
(7) The interrupt flag is cleared, unless the peripheral has more than one type of interrupt.
(8) The SR is cleared, which disables any further interrupts during the ISR handling and wakes up the processor from any low power mode.
    - Only "maskable" interrupts are disabled by the SR clear, this means "non maskable" interrupts, such as reset, are still possible during an ISR.
(9) The address of the interrupt vecor is copied to the PC, directing the processor to start executing code from that address on.

Every ISR requires some special syntax:

```
#pragma vector = <VECTOR_NAME>
__interrupt void <ISR_NAME> (void){
        // ISR Code
    }
```

Conclusion

The whole experience of working at Capital Power Systems Ltd. was great. This company has good work culture and very high quality of work ethics. I learned a lot about the technology used nowadays in the dynamic market of electronics goods and services. The work I could complete here was very satisfactory. I have tried to develop as many add-ons in my skill set as possible. I hope my work at this company help it meet its goals.

References

(1) http://www.ti.com/lsds/ti/microcontrollers-16-bit-32-bit/msp/overview.page
(2) http://www.electronics-tutorials.ws/
(3) https://community.nxp.com/docs/DOC-1091
(4) https://learn.sparkfun.com/tutorials/pcb-basics
(5) http://www.metertocash.com/easy-understand-overview-electric-meter.html
(6) https://en.wikipedia.org/wiki/Smart_meter
(7) MSP430 microcontroller basics by John Davies.