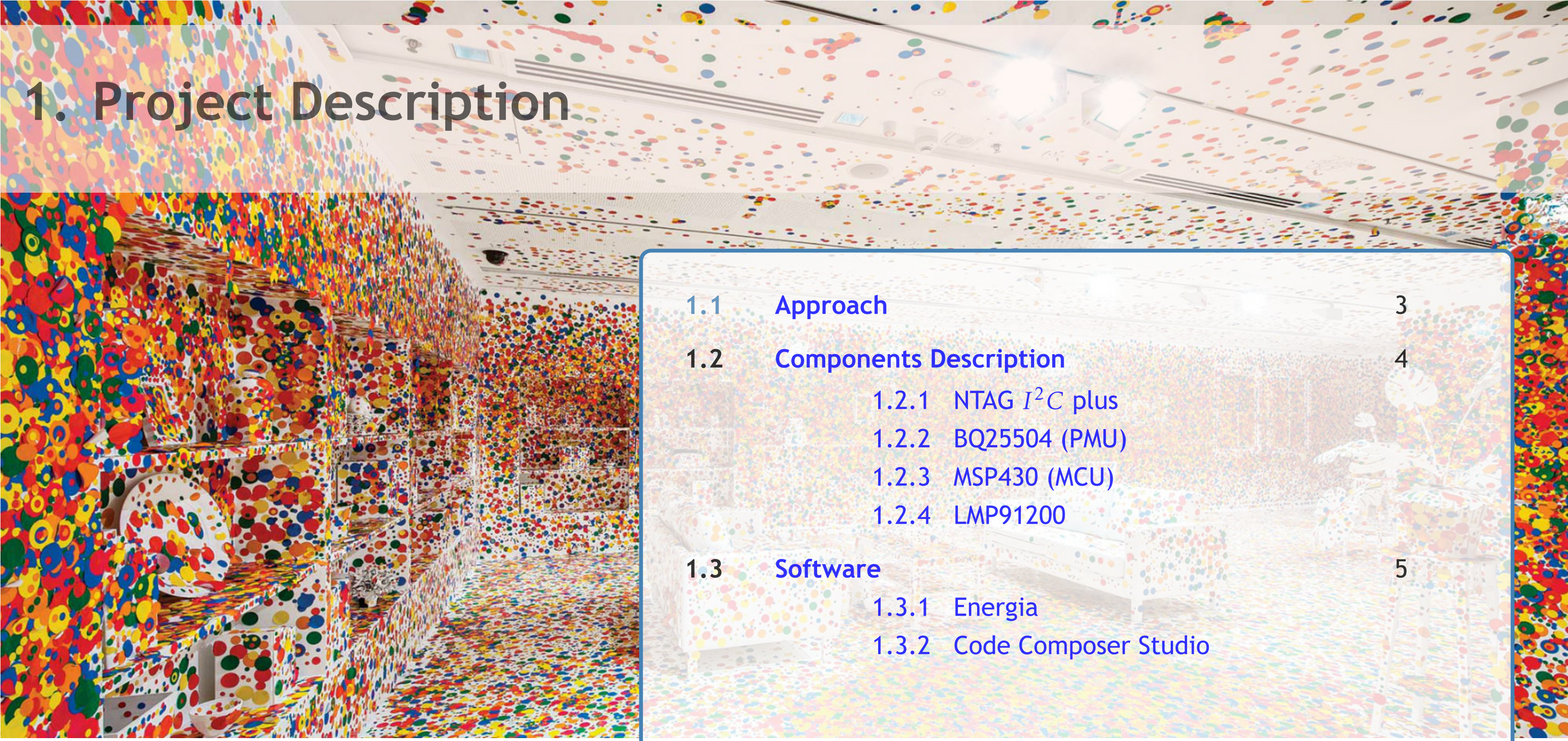
Battery-less Water Pollution Monitor
EE 344 - Electronic Design Lab

Mihir Kavishwar - 17D070004
Rishabh Dahale - 17D070008
Mithilesh Vaidya - 17D070011

Guide: Prof. Siddhart Tallur

Contents



1. Project Description

This project aims at developing a generic platform consisting of a micro-controller and other electronic components powered using Near Field Communication (NFC) based energy-harvesting techniques. This platform can be interfaced with any passive or low power analog sensor, which will be used to read the parameter of interest and communicate the value back to a NFC device (e.g. a smartphone) using the NFC antenna embedded in the device.

1.1 Approach

To get a better understanding of how NFC works and it's limitations, we initially worked with a production-ready NFC IC and antenna. For this, we used NXP's NTAG I^2C plus Explorer Kit. It contains a class 4 antenna and a class 6 flex antenna. This kit helped us understand the basic of NFC communication.
After a detailed study, we found that traditional micro-controllers could not be used for this application as they consume a lot of power. Details about various micro-controllers are given in the following section.
For powering up the circuit, we initially tried to connect the NXP's NFC energy harvesting IC directly to the micro-controller, but there were constant variations in the current provided by NFC which led to unpredictable circuit behaviour. To overcome this, we used TI's BQ25504. It is a ultra low power DC-DC boost converter. It could provide stable output for powering our circuit.
For the purpose of demonstration, we would be measuring the pH of water. For this: we use an electrode probe along with TI's LMP91200 configurable Analog front-end (AFE).

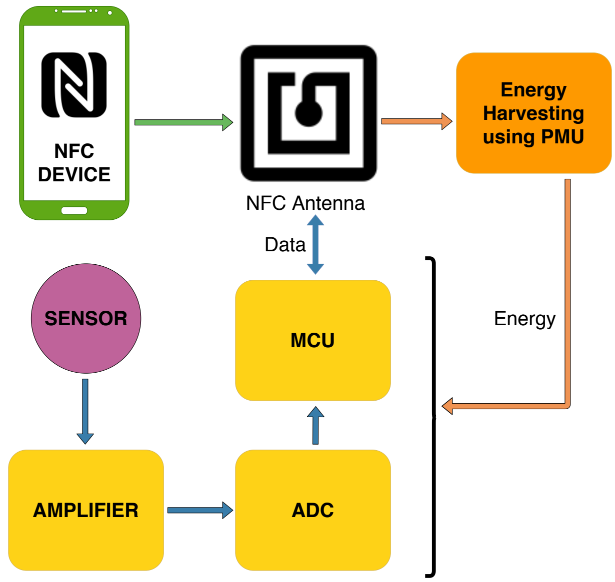


Figure 1.1: High-level Flowchart of the approach

1.2 Components Description

1.2.1 NTAG I^2C plus

NXP NTAG I2C plus is a family of connected NFC tags that combine a passive NFC interface with a contact I2C interface. It features 2kB of EEPROM and – 64 bytes SRAM buffer for transfer of data between NFC and I2C interfaces. It also does energy harvesting from the NFC field. The I^2C slave supports 7-bit slave addressing and supports frequencies upto 400kHz. The energy harvesting feature can deliver upto 15mW of power from the NFC field.

1.2.2 BQ25504 (PMU)

The BQ25504 IC is an intelligent integrated energy harvesting nano-power management solution that is well suited for meeting the special needs of ultra-low power applications. The device is specifically designed to efficiently acquire and manage power in the range of Microwatts (μW) to a few Milliwatts (mW).

The design of the BQ25504 starts with a DC-DC boost converter/charger that requires few Microwatts to begin operating. Once powered up, the boost converter/charger can effectively extract power from low-voltage output harvesters. The boost converter can be started with V_{in} as low as 600 mV, and once started, can continue to harvest energy with an input of just $V_{in} = 130mV$. The above design parameters of BQ makes it as an ideal candidate for our low power NFC application.

1.2.3 MSP430 (MCU)

The MSP430G2x13 and MSP430G2x53 series are ultra-low-power mixed signal micro-controllers. For our project, we selected MSP430G2553 since it was already available in WEL and met our power requirements. It has a flash memory of 16kB and a 512-byte RAM. The device have an active current of 230μA at 1MHz, 2.2V, whereas a standby current of 0.5μA. The device supports input voltage range of 1.8V to 3.6V. It has a 16-bit RISC architecture, and also supports Universal Serial Communication Interface. It has an inbuilt ADC.

1.2.4 LMP91200

The LMP91200 device is a sensor AFE for use in low-power, analytical-sensing applications. The LMP91200 is designed for 2-electrode sensors. This device provides all of the functionality needed to detect changes based on a delta voltage at the sensor. Optimized for low-power applications, the LMP91200 works over a voltage range of 1.8 V to 5.5 V. With its extremely low input bias current, it is optimized for use with pH sensors. Also, in absence of supply voltage the very low input bias current reduces degradation of the pH probe when connected to the LMP91200. Two guard pins provide support for high parasitic impedance wiring. Depending on the configuration, total current consumption for the device is 50 μA while measuring pH.

1.3 Software

The project requires the use of micro-controller. It needs to be programmed to perform the communication between the sensor and the NFC IC. Texas Instruments provides two softwares which can can be used to program the MSP430:

1.3.1 Energia

Energia is an open source community-driven integrated development environment (IDE) software framework. Based on the Wiring framework, Energia provides an intuitive coding environment as well as a robust framework of easy-to-use functional APIs libraries for programming a micro-controller.

Advantages

1. Simple, easy-to-use code editor compiler with built-in Serial Monitor/terminal
2. Features a robust framework of intuitive functional APIs for controlling micro-controller peripherals (i.e. digitalRead, digitalWrite, Serial.print, etc)
3. Support for various TI embedded devices (MSP430, TM4C, CC3200, C2000, etc)
4. Higher level libraries are also available (i.e. Wi-Fi, Ethernet, displays, sensors more)

Disadvantage

Due to a heavy dependence on the in-built libraries and API, the user does not have a register-level control of the micro-controller. For our project, we need to ensure that the micro-controller utilizes bare minimum power as we have limited energy supply coming from the NFC antenna. MSP430 has a watchdog timer. It's main job is to reset the micro-controller when it gets trapped in an infinite loop or experiences a software fault. It is basically a hardware timer which counts up and resets the controller after the specified time has passed. Energia uses the Watchdog timer in timer mode for it's internal use. But this timer utilizes a lot of power.

Due to this excess power consumption, we were not able to power up the MSP using energy harvested from NFC.
This was the main reason for shifting from Energia to CCS.

1.3.2 Code Composer Studio

Code Composer Studio (CCS) is an integrated development environment (IDE) that supports TI's Microcontroller and Embedded Processors portfolio. Code Composer Studio comprises a suite of tools used to develop and debug embedded applications. It includes a C/C++ compiler with optimization capabilities, a source code editor, project build environment, debugger, profiler, and many other features. The intuitive IDE provides a single user interface taking you through each step of the application development flow.

Firstly, we switched off the Watchdog timer since it consumed excess power. Then, we programmed it do simple tasks such as blink an LED to better understand the registers and ports on MSP430. Once we were comfortable with programming the MSP430, we read the I2C documentation for MSP430.

NXP NTAG I2C provides an Android app through which one can send and receive data. Our first experiment was to send a string from MSP430 to the NTAG IC and verify whether the communication was successful by reading the registers of the NTAG IC using a smartphone.

The MSP430 acts as the master since it provides the clock while the NTAG I2C acts as the slave. Note that this is just for testing purposes. In the final project, the NFC IC will acts as the master sinnce it will be generating the clock. Also, the MSP430 is the transmitter since it sends a string to the NTAG I2C.

Given below are the steps for initialising the registers for transmitting data from MSP430 to the NTAG IC via I2C:

- Set the UCSWRST bit for setting the parameters.
- Set UCMODE0 = 11, USYNC = 1 and set the UCMST bit to set the MSP430 as the master.
- Clear the UCSWRST bit to complete the configuration.
- Store the slave address (0x55 in case of NTAG) in the register UCB0I2CSA.
- Since we are using 7-bit addressing mode, clear the UCSLA10 bit.
- Set the UCTR bit since the MSP430 will be transmitting data.
- Set UCTXSTT to generate a start condition. The slave address is also sent on the I2C line and an acknowledgment is received by the MSP430 from the NTAG IC. UCB0TXIFG bit is then set.
- Data to be transmitted (a string in our case) is moved into the UCB0TXBUF register. After the transfer is complete, UCB0TXIFG is set again. We can then send the next byte.

In this way, we transfer a string from MSP430 to NTAG and use a smartphone to read the NTAG registers. The main issue we faced was configuring the NTAG IC so that the incoming data from MSP430 could be dumped at the desired address. Data would get dumped at random locations in the NTAG IC memory. We were not able to debug this issue. Given below is a screenshot of the NTAG IC memory. The string 'EDL' is dumped at address 006.

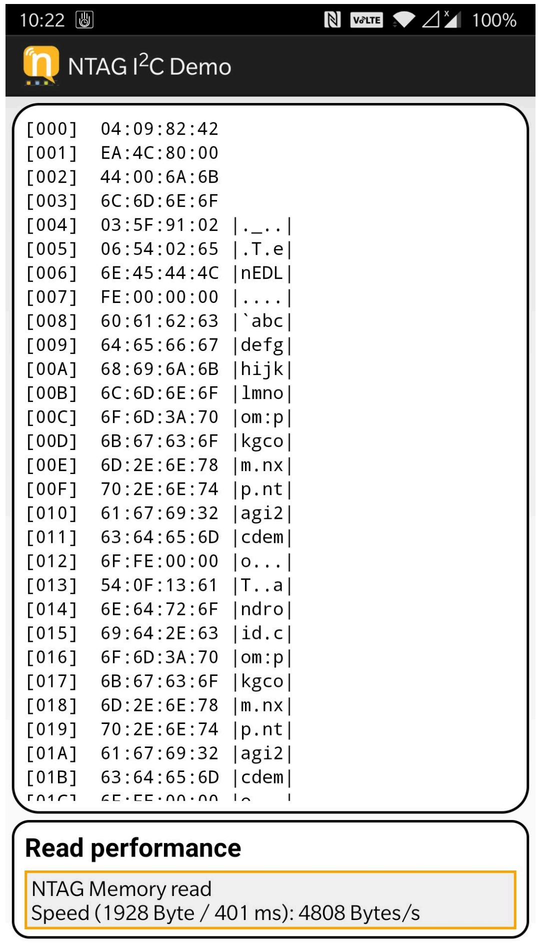
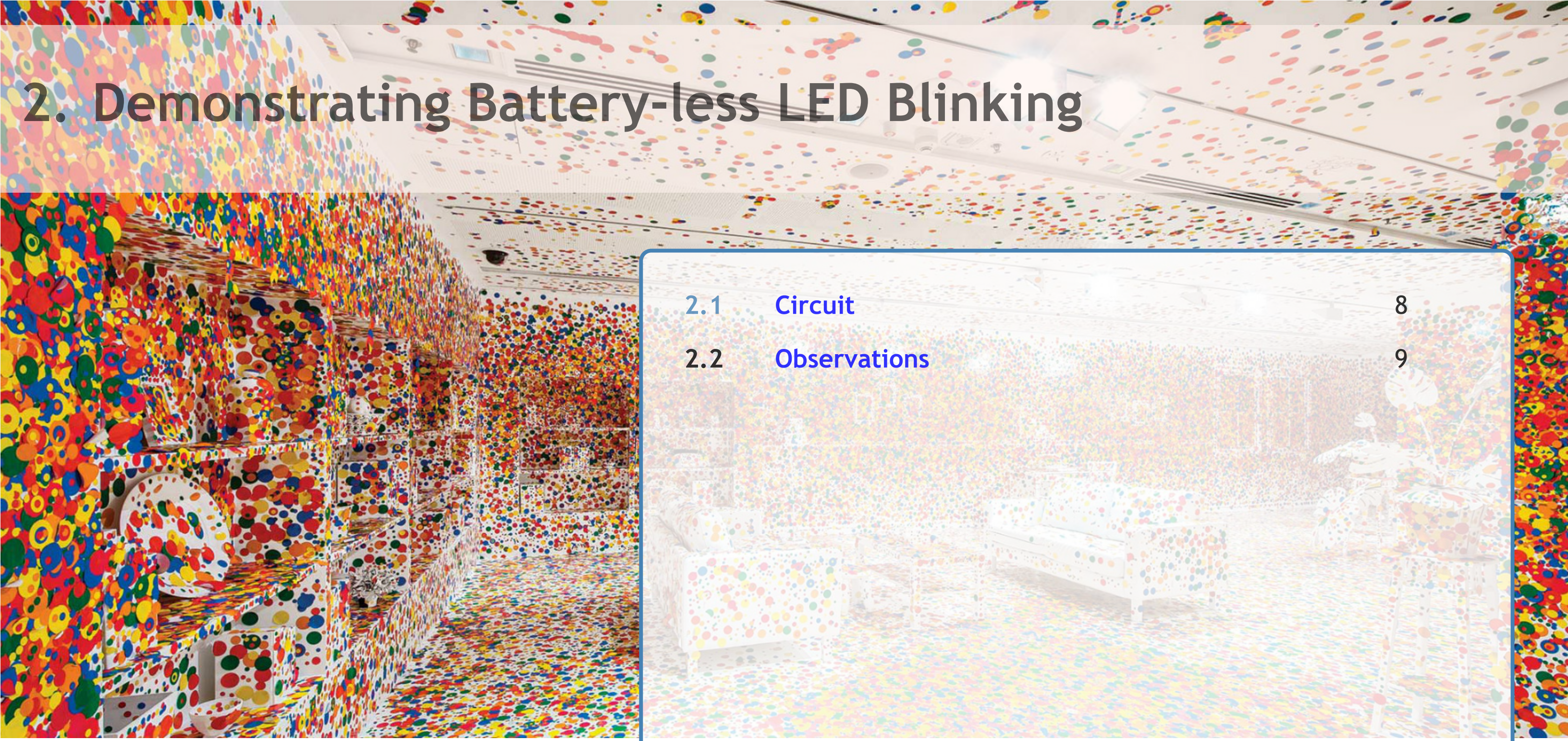


Figure 1.2: NTAG memory



2. Demonstrating Battery-less LED Blinking

Analogous to printing "Hello World" while learning a new programming language, blinking an LED is the first step while working in the world of embedded systems. The challenging part in our project is to do this without a battery or external power source and instead use NFC based energy harvesting. After a few failed attempts, we were able to successfully demonstrate a blinking LED.

2.1 Circuit

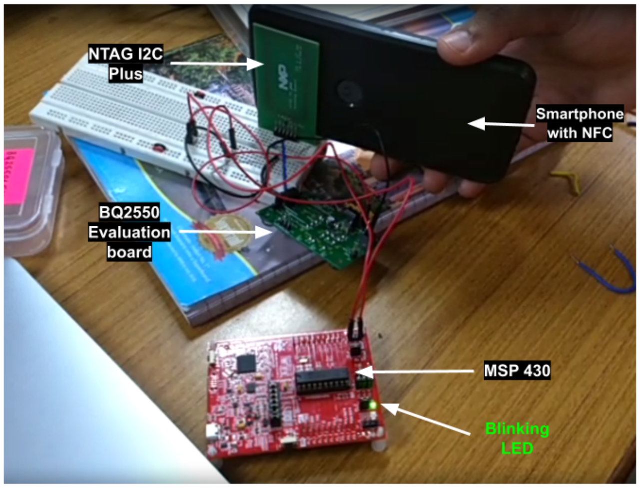


Figure 2.1: Demonstration of LED Blinking

Figure 2.1 shows the basic setup of NTAG and MSP430. The initial experimenting was done with a breadboard. To make the circuit more reliable we later switched to a perforated board as shown in figure 2.2.

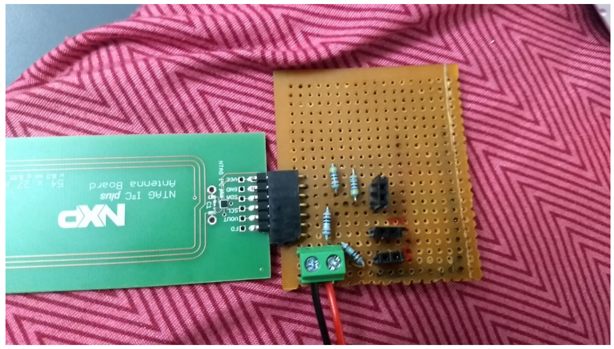
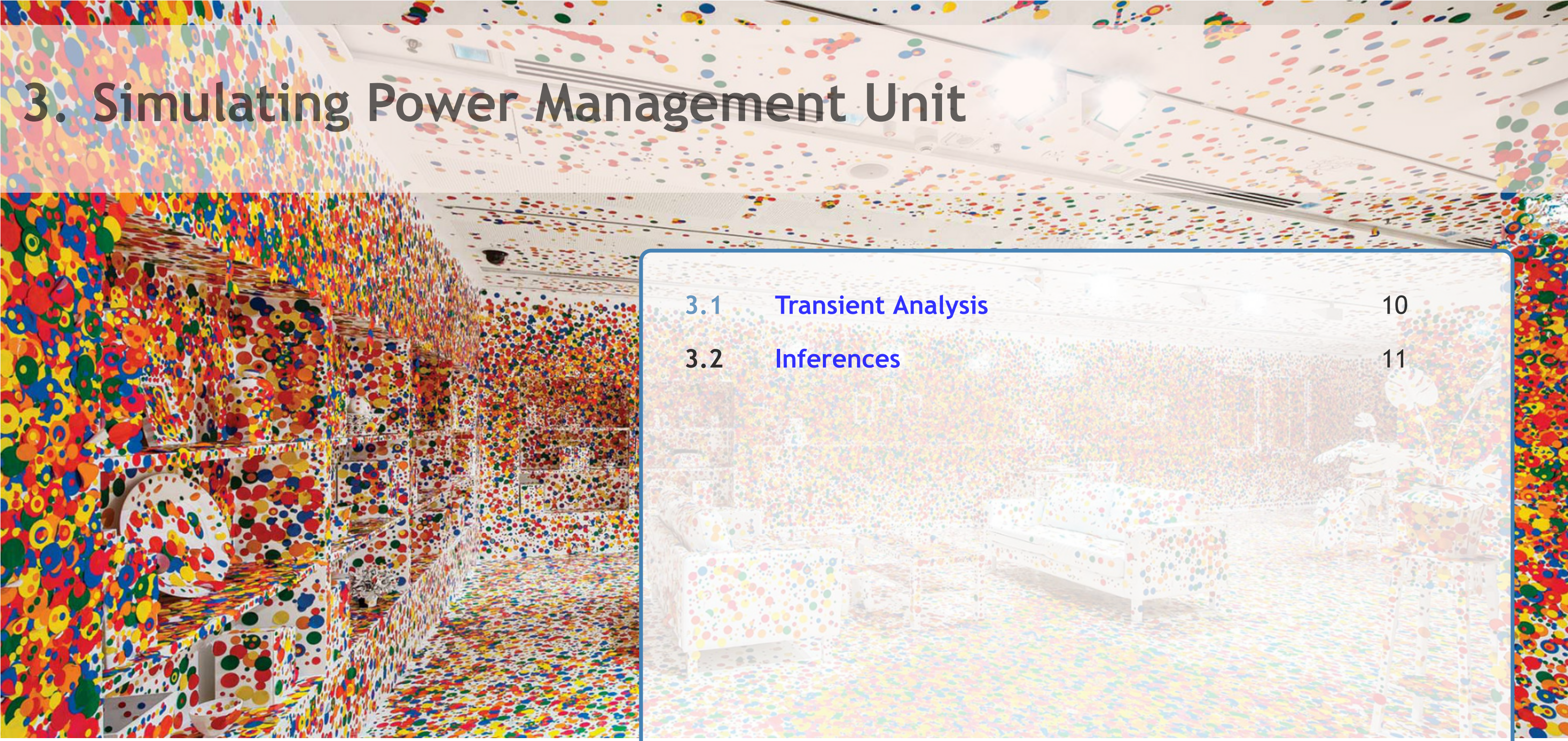


Figure 2.2: Ciruit on perforated board

2.2 Observations

1. A voltage of around 3.3 V was generated at the output of NTAG IC when the smartphone was at a distance of nearly 1 cm.
2. Connections to serial interface (UART) had to be disabled on the micro-controller to make it operate in low power mode so that the power requirements were within what NFC could deliver.
3. Energy could also be harvested if there was a 2 mm thick plastic barrier in between the antenna and source. However, the brightness of LED was reduced.
4. There were reliability issues and sometimes power fluctuations triggered watchdog timer of the MCU causing a hardware reset.



3. Simulating Power Management Unit

In order to design a good system, it is always a good practice to carry out simulations and analyze the results. Since Power Management Unit is responsible for maintaining a constant supply voltage to the entire circuit, we decided to simulate its transient behavior. We used the software: Toolkit for Interactive Network Analysis - Texas Instruments (TINA-TI) to carry out these simulations.

3.1 Transient Analysis

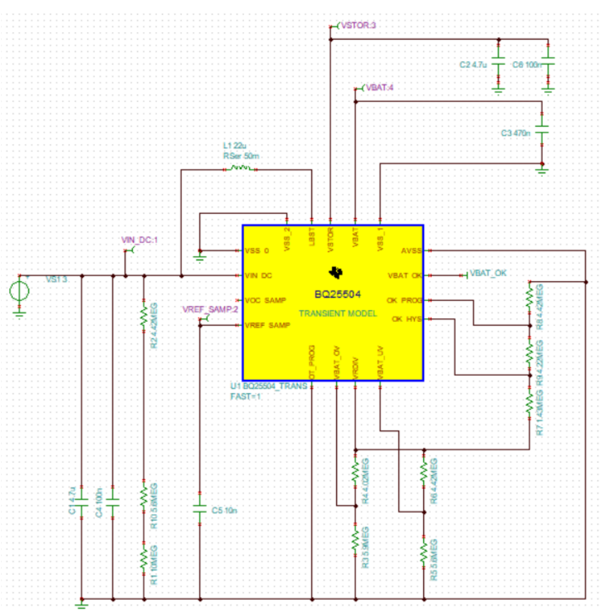


Figure 3.1: PMU Schematic on TINA-TI

Texas Instruments provides TINA SPICE models for all of it's ICs. We used the model for BQ25504 to create this schematic and then perform simulations. Since it is difficult to calculate the exact voltage generated at the input of PMU due to NFC, we instead placed a Voltage source with some random noise at the input. This may not be the exact physical model of our system but does the job of a behavioral model.

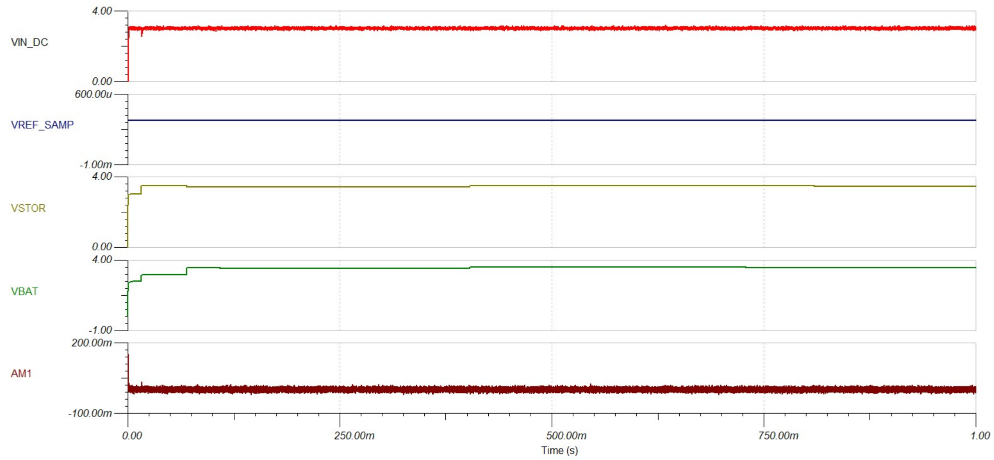


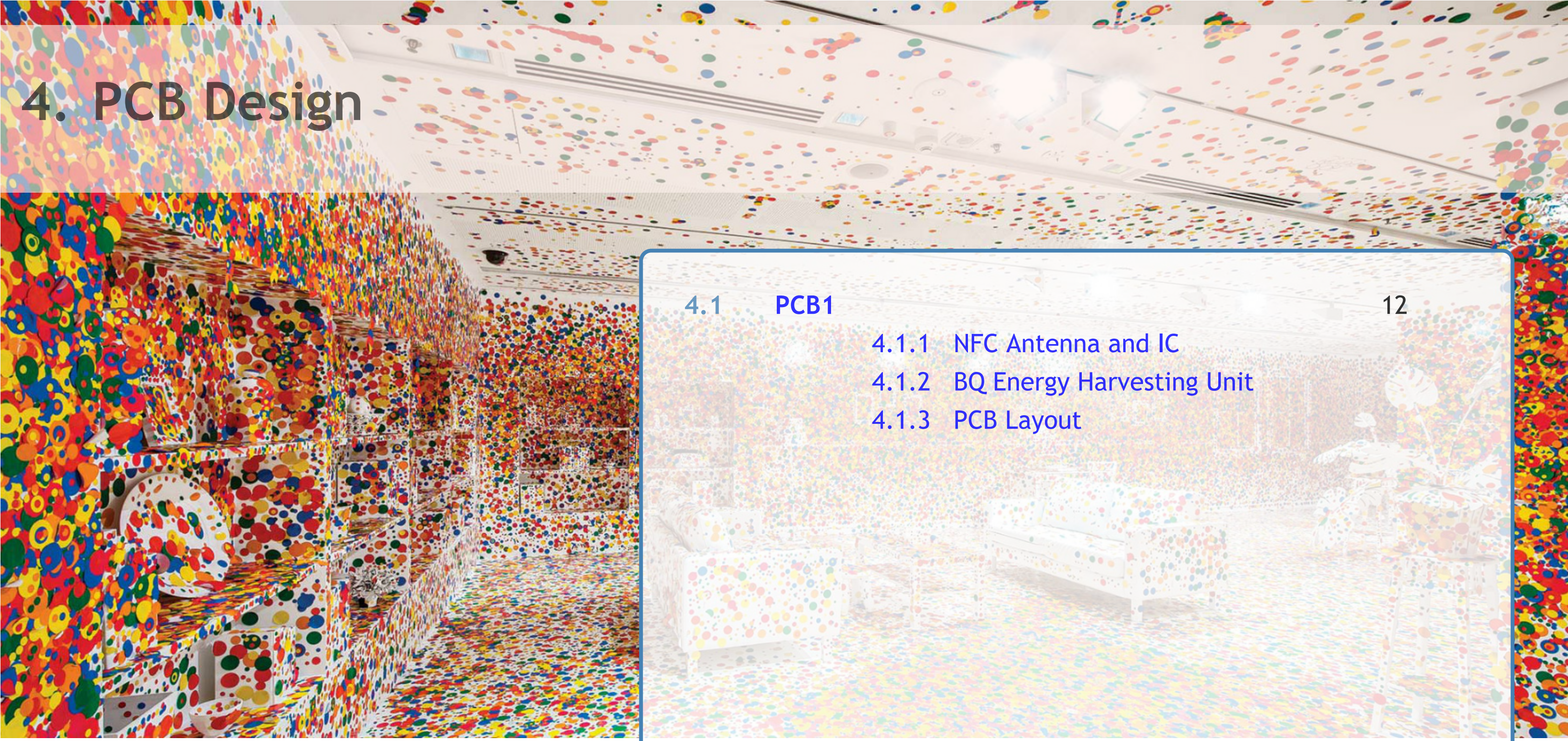
Figure 3.2: Transient Analysis of PMU in TINA-TI

For simulation purposes, we set the input voltage $VIN\_DC = 3.3V$ with some additive White Gaussian noise. This value was chosen after observing the actual voltage we got when antenna was kept at a distance of 1 cm from the NFC reader.

At the output (VSTOR), we have a capacitive load. The figure 3.2 indicates that the output fluctuates for few milliseconds before reaching a nearly stable value of 3.4 V. This is despite the variation on input voltage. The steady state current requirements are not very clear but it seems to be in the order of few mA. Startup current is nearly 140 mA.

3.2 Inferences

1. Even if there are fluctuations of $\pm 1 mV$ in voltage generated from NFC energy harvesting, the output voltage of PMU should be stable.
2. The PMU circuit has some kind of boost converter because of which the output voltage is slightly greater than the input voltage.
3. Starting current is much higher that the steady state current. This might give rise to some kind of feedback causing a voltage drop at the input but that hasn't been modelled here.



4. PCB Design

As we planned to develop a generic NFC-based platform for low power measurements, we planned to develop 2 PCBs. One containing the NFC antenna and the power management unit, while the other containing the micro-controller and it's peripherals for the measurements. This will give us the advantage of pairing the same communication board with different controllers/peripherals for different measurements.

4.1 PCB1

4.1.1 NFC Antenna and IC

The NFC antenna is a core element of our project since it extracts power from the NFC field generated by a smartphone. Details of the antenna design should be checked in each IC application note as there are some consideration to take into account for each of them. For our application, we used NXP's class 4 antenna design. It is a 5-turn, 50pF antenna with a size of about $48 \times 25mm$. We chose this design over class 6 antenna because of the size. Our primary goal is to use this to communicate with mobile phone, and class 6 antenna were

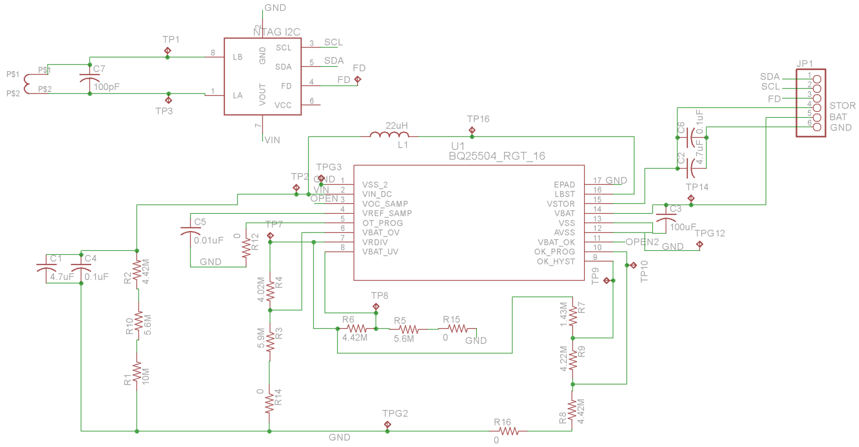


Figure 4.1: PCB1 design schematic

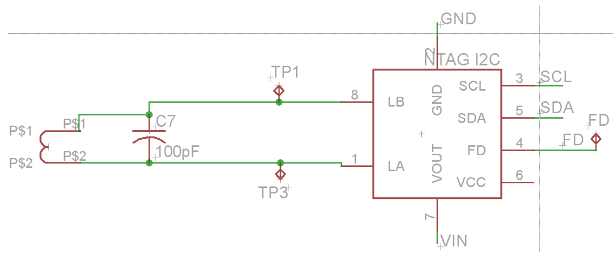


Figure 4.2: NFC antenna-IC interface schematic

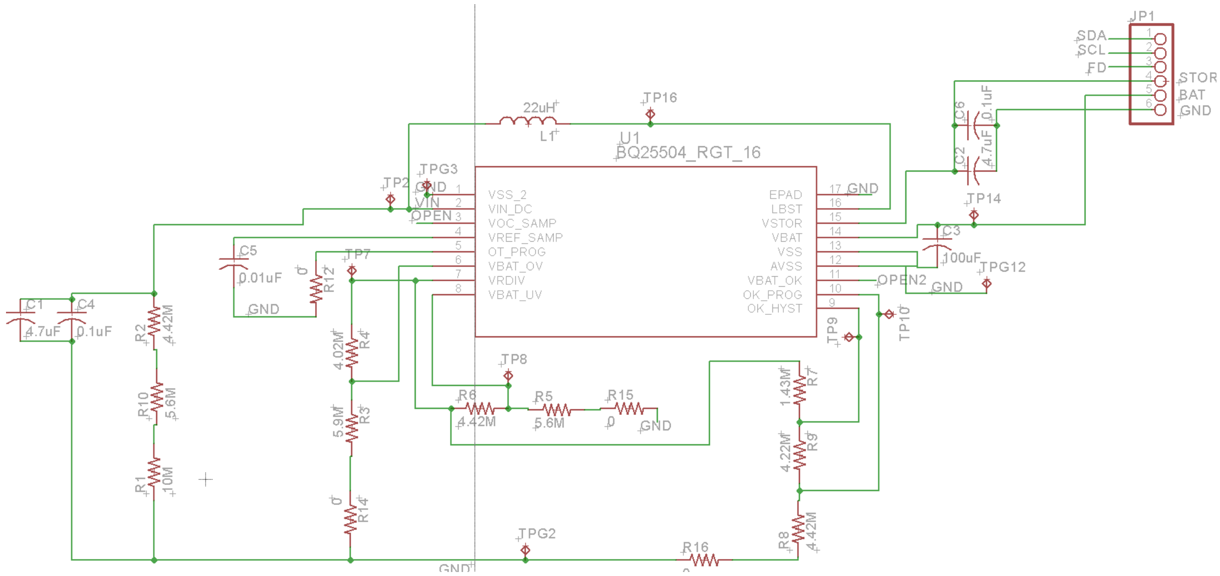


Figure 4.3: BQ IC schematic

smaller in size making it difficult to properly orient the phone for optimum energy harvesting. The circuit schematic of this interface is shown in figure 4.2.

4.1.2 BQ Energy Harvesting Unit

While BQ25504 provides a lot of features apart from effectively acquiring and managing μW to mW of power, like programmable dynamic maximum power point tracking (MPPT), ultra-low-power with high-efficiency DC/DC boost converter/charger, energy storage, we do not need all the functionalities. To cater to our needs we added the circuits for this unit in our PCB1. The circuit schematic for the same is shown in figure 4.3.
The jumper pins on right side of figure 4.3 were to be used for connecting to the second PCB which contains the micro-controller and it's peripherals.

4.1.3 PCB Layout

[h] The final layout of our PCB after some iterations and help from Maheshwar sir is shown in figure 4.4.

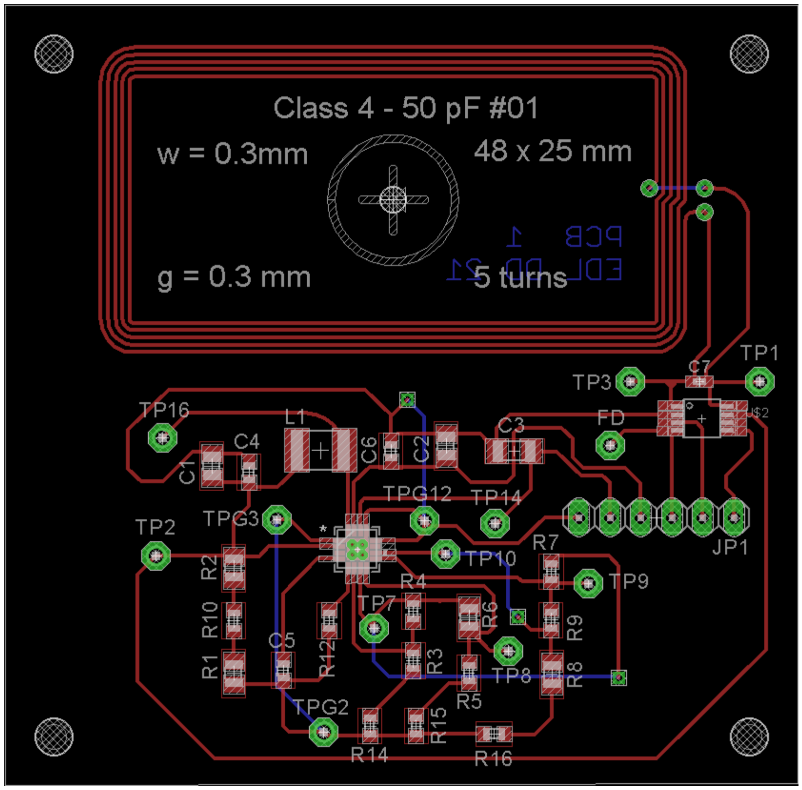


Figure 4.4: PCB1 Layout

We had started working on the schematic of **PCB2**, but due to the bitter turn of events, couldn't complete it :(