Subassembly Specifications Report

Team 3

Date: 09.12.2011

Authors:

Oğuzhan Avcı (HW Engineer)

Abdullah Başar Akbay (HW Engineer)

Version I

Table of Contents

1. GENERAL INFORMATION

1.1 Software Structures

Software is run on smart phone. Its responsibility is providing a user interface, obtaining GPS coordinates from the GPS chip, drive D/E block for encryption of the transmitted data and decryption of the received data and organise the user network. Its relationship with the D/E block is described in detail in Preliminary Design report. Some of the communication protocols between D/E Block and phone are determined and explained again in Preliminary Design Report.

1.2 User Interfaces

The device will be paired with a smartphone. There will be a button on the device for turning it on/off, so it can be seen by the smartphones around via Bluetooth. For simplicity and easy to use purposes, the device will only have one button. After successfully pairing the device, when the application is used on the smartphone, the device automatically receives the location data, encrypts it and transmits it back to the smartphone.

2.MAIN BLOCKS

2.1 System Block Diagram

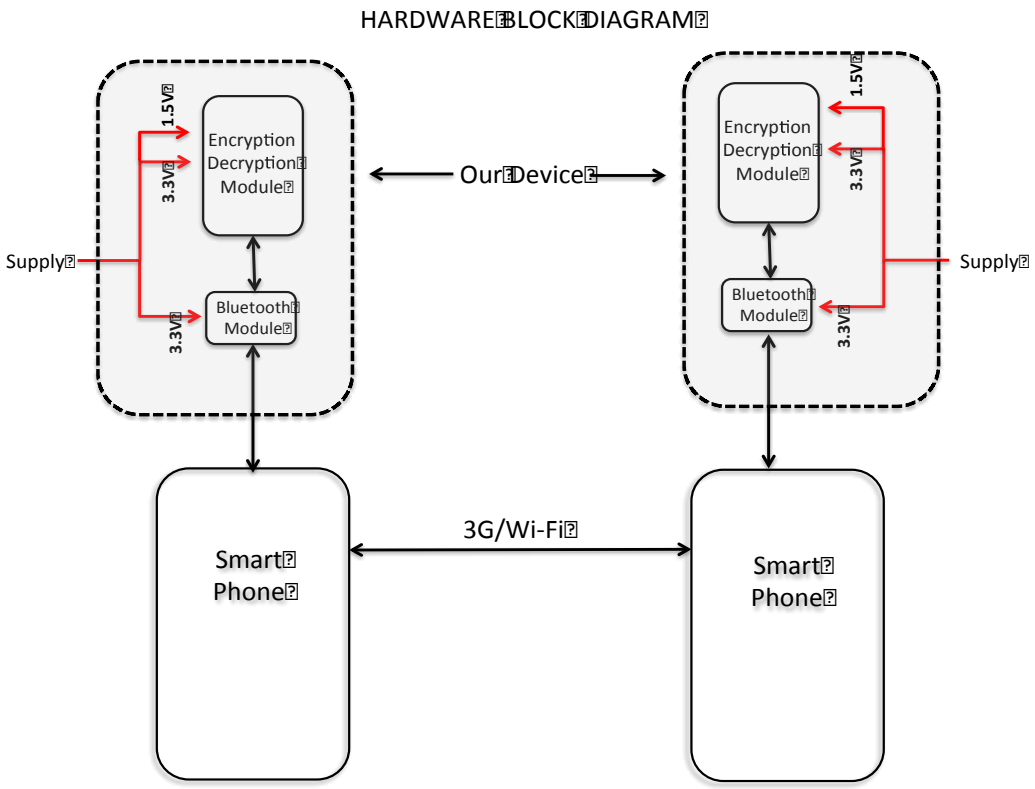


Figure 1: System Block Diagram

2.2 Product Tree

2.2.1 Product Tree of SCC

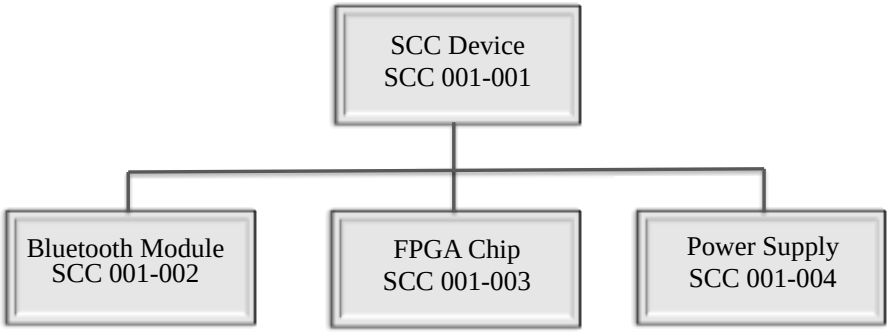


Figure 2: Product Tree

2.2.2 Subassembly Product Tree

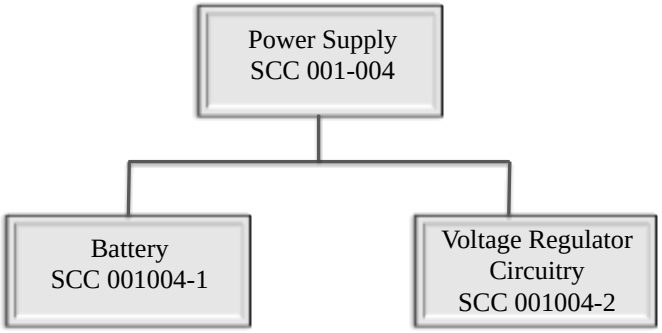


Figure 3: Subassembly Product Tree

3. SUBASSEMBLY SPECIFICATIONS

3.1 SCC 001002: Bluetooth Module

We will use Texas Instruments' CC2540 Bluetooth chip, which is shown in figure 2. The transceiver block receives the information, which is the location of the user and message and sends it to the encryption block. The connection of D/E and Smartphone is done via this block.

The specifications of the CC2540 are as follows: [1]

- Dimensions: 6-mm × 6-mm Package
- Operating voltage: 2V-3.6V
- Current Consumptions:

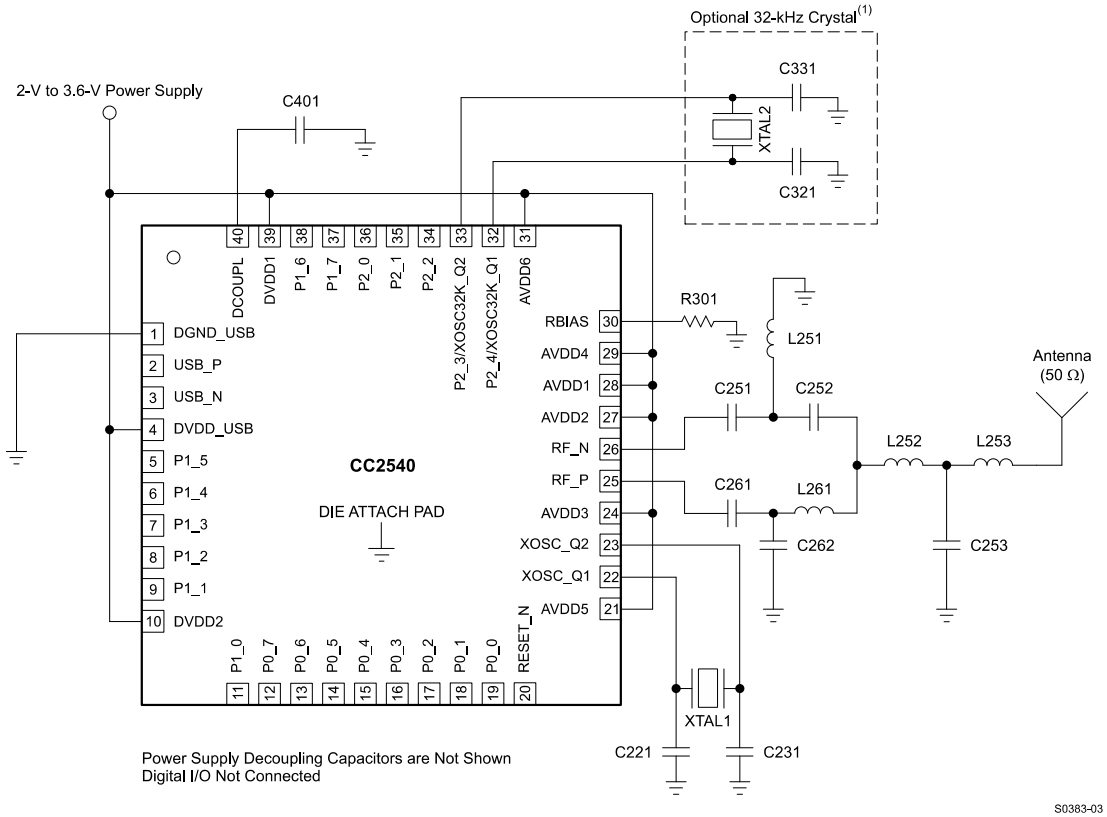
  Active Mode RX Down to 19.6 mA

  Active Mode TX (–6 dBm): 24 mA

  Power Mode 1 (3-µs Wake-Up): 235 µA

  Power Mode 2 (Sleep Timer On): 0.9 µA

  Power Mode 3 (External Interrupts): 0.4 µA
- Operating Temperature: -40^{o}C – 85^{o}C
- Operating Frequency: 4MHz (Master&Slave RX and TX)



(1) 32-kHz crystal is mandatory when running the chip in low-power modes, except if the link layer is in the standby state (Vol. 6 Part B Section 1.1 in [1]).

NOTE: Different antenna alternatives will be provided as reference designs.

Figure 4: CC2540 Application Circuitry [1]

Figure 5: CC2540 Texas Ins.

3.2 SCC 001003: FPGA Chip

3.2.1 Overview of SCC 001003: FPGA Chip

FPGA Chip is used to implement D/E Block which is responsible for the decryption and encryption of the data. In Product Specification Document, under Functional and Performance Specifications heading it has been expressed that the encryption method must be Public Key Cryptography. Furthermore, it has been added that it should use a key with sufficiently higher number of bits. The minimum length of the key is determined as 128 bits. ALTERA EP1C3T100C8N, which has 2910 logic elements, will be used to implement the encryption/decryption algorithm. The FPGA chip has the following specifications: [2]

- Supply voltage (V_{CCINT}): -0.5V-2.4V
- DC Input Voltage: -0.5V-4.6V
- Output supply voltage: 3.0V-3.6V
- High-level input voltage: 1.7V-4.1V
- Low-level input voltage: -0.5V-0.7V
- Supply voltage for internal logic and input buffers: 1.425V-1.575V
- Supply voltage for output buffers, 3.3-V operation: 3.0V-3.6V
- Storage temperature: -65^{o}C-150^{o}C
- Ambient temperature: -65^{o}C-135^{o}C
- Junction temperature: 135^{o}C(max)
- Dimensions: 16mm x 16mm (WxH)

In Preliminary Design Report, objectives of the D/E Block are explained in detail on pages 7-10. Its relationship with the other main blocks and interfaces are also expressed in the minutest detail. In this Subassembly Specifications Report, internal working and operations of the D/E Block are presented.

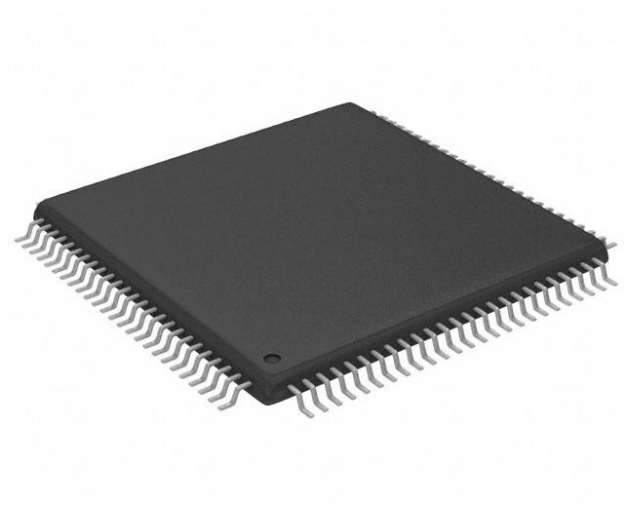


Figure 6: ALTERA EP1C3T100C8N [2]

3.2.2 Internal Design of SCC 001003: FPGA Chip

Detailed logical block diagram of D/E Block is given on Figure 7. It is an improved version of internal block diagram given in Block Diagram & Flowchart Report. Note that only logical operations are shown on this figure. One deficiency of this block diagram is that it does not satisfy the package communication design defined in the Preliminary Design Report. The plaintext and ciphertext are sent as 10 bytes packages. In other words, a buffer should be implemented within the FPGA to store 10 bytes information temporarily. This will be added to the block diagram in the further designs.

Descriptions of the submodules are given below:

- SWITCH Module: Bluetooth Module has 8 I/O pins. FPGA must use the same pins for sending and receiving data. Switch module decides whether data is transmitting or receiving based on the condition inside the process inside FPGA. This information comes from the OUTPUT Module.

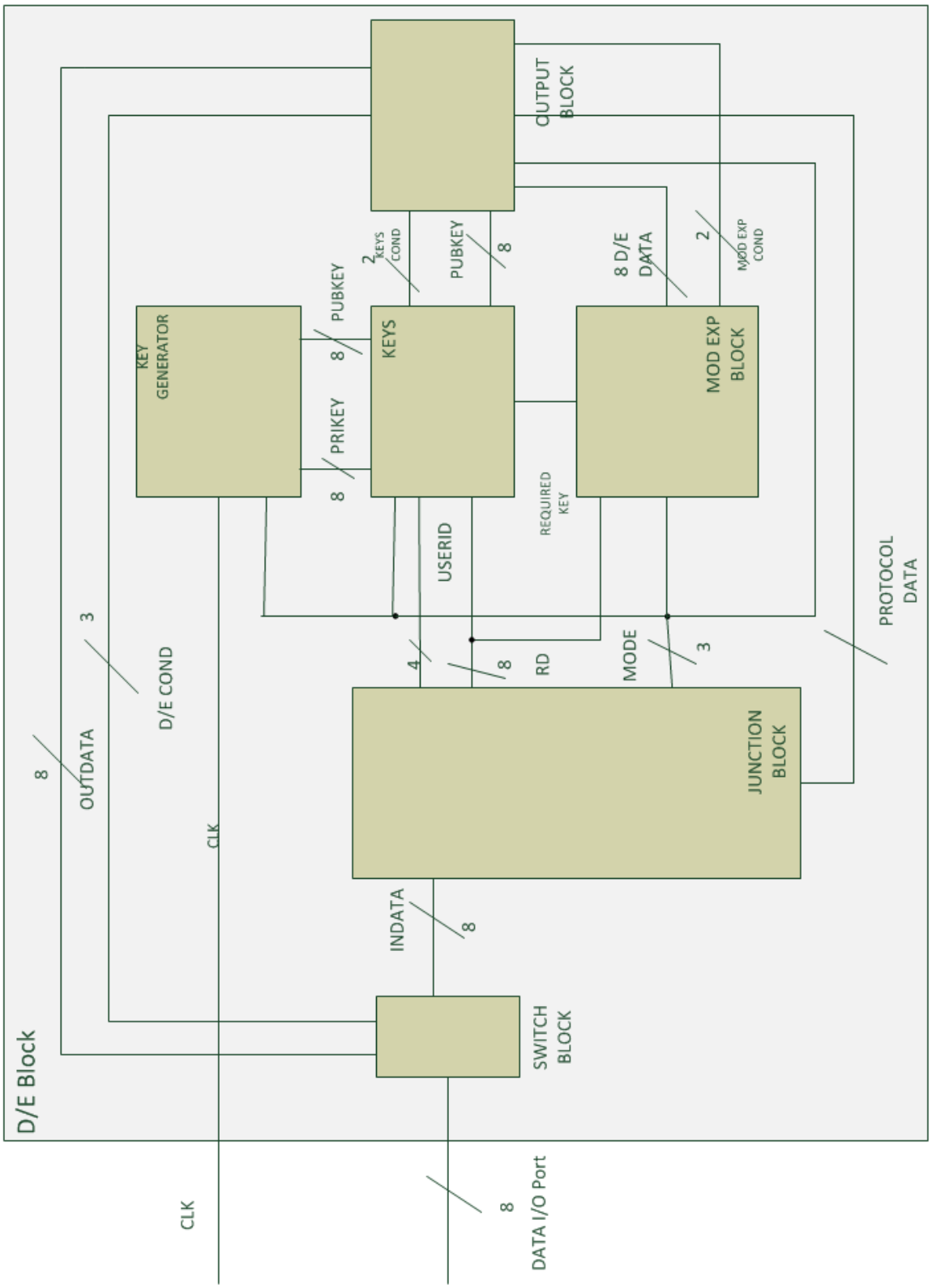


Figure 7 – Logical Block Diagram of D/E Block

- JUNCTION Module: This module interprets the incoming message from the phone. Its outputs are 8 bit vectors of RD and PROTOCOLDATA, and 3 bit vector MODE. MODE is the input to KEYGENERATOR, KEYS and MODEEXP submodules. MODE carries the information of the current activity of the D/E Block such as encryption, decryption, public key sending, public key receiving, key generation and standby. If the incoming data will be used in encryption, decryption or public key storage, it has been outputted on RD 8-bits vector bus. If D/E block is implementing a protocol, JUNCTION Module prepares the answer and sends it to the OUTPUT Module. JUNCTION also indicates the user identity to KEYS module when the message is encrypted using this user's public key to be sent to this user.
- KEYS Module stores the public, private keys of the user and public keys of other users. It takes MODE and USERID signals from JUNCTION module. Based on this information it sends the required keys to MODEXP Block. KEYGENERATOR Module also sends its outputs which are PUBKEY and PRIKEY to KEYS module.
- MODEXP Block performs modular exponentiation. Result of this function is plaintext or ciphertext depending upon which key and data is used.
- KEYGENERATOR Module: KEYGENERATOR Module implements the RSA Key Generation Algorithm which is given below. It gives two outputs which are public key and private key. KEYGENERATOR module is activated when the user is log on. This information is indicated by the phone and JUNCTION Module arranges MODE signal accordingly.

  RSA Key generation algorithm is given below:

"Generate two large random primes, p and q, of approximately equal size such that their product n = pq is of the required bit length, e.g. 1024 bits.

Compute n = pq and (phi) φ = (p-1)(q-1).

Choose an integer e, 1 < e < phi, such that gcd(e, phi) = 1.

Compute the secret exponent d, 1 < d < phi, such that ed ≡ 1 (mod phi).

The public key is (n, e) and the private key (d, p, q). Keep all the values d, p, q and phi secret. [We prefer sometimes to write the private key as (n, d) because you need the value of n when using d.]

n is known as the modulus.

e is known as the public exponent or encryption exponent or just the exponent.

d is known as the secret exponent or decryption exponent." [3]

KEYGENERATOR Module has also submodules:

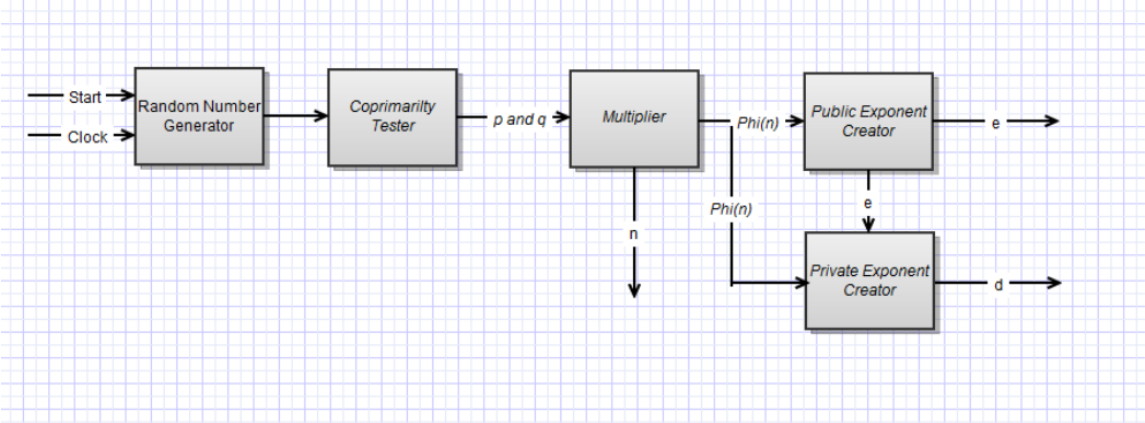


Figure 8 – KEYGENERATOR Module

Further details on the implementation of the KEYGENERATOR Module are given in the Appendix.

3.3 SCC 001004: Power Supply

This part consists of 2 subassemblies: Battery and Voltage Regulator Circuitry. The battery will supply the 1.5V and 3.3V levels for the FPGA and Bluetooth chips. The voltage regulator circuitry will make sure the FPGA gets the appropriate voltage levels due to the fact that the logic device is sensitive to the voltage and current variations.

3.3.1 SCC 001004-1: Battery

For power supply, we are going to use SANYO Li-ion Prismatic (UF103450P) 3.7V 2000mAh with Fuse battery, which will supply 3.3V and 5V to the Bluetooth and FPGA chips. The battery has the following specifications: [3]

- Nominal Voltage: 3.7V
- Nominal Capacity: Min. 1880mAh
- Charging Voltage: 4.2V
- Ambient Temperature:

  Charge: $0^{o}C$ – $40^{o}C$

  Discharge: $-20^{o}C$ – $60^{o}C$

  Storage: $-20^{o}C$ – $50^{o}C$
- Charging Time: 2.5hrs
- Weight: 38.5g
- Dimensions: 10.60mm x 33.90mm x 48.80mm (TxWxH)

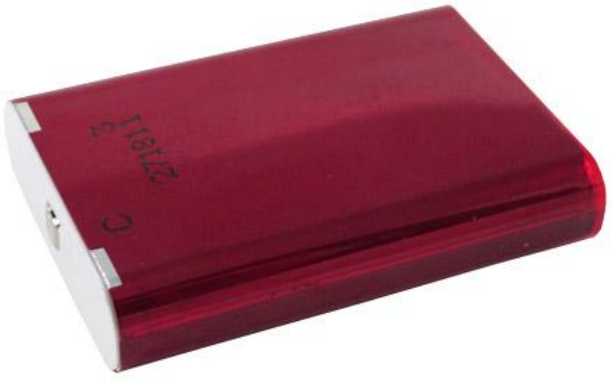


Figure 9: SANYO Li-ion Prismatic (UF103450P) Battery [4]

3.3.2 SCC 001004-2: Voltage Regulator Circuitry

This part is responsible for FPGA getting appropriate voltages at its pins, 1.5V and 3.3V. Thus, in the voltage regulator circuitry, there will be 2 types of regulators: one for 1.5V and another for 3.3V. LF15AB type regulator will be used to ensure 1.5V level, which has the following specifications: [5]

- Output voltage: 1.485V-1.515V
- Operating input voltage: -2.5V-16V
- Output noise voltage: 50uV
- Control input logic low: 0.8V
- Control input logic high: 2V
- Temperature range: -40^{o}C – 125^{o}C

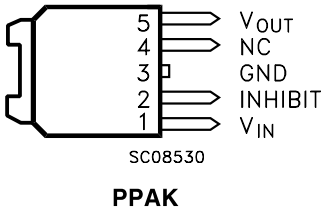


Figure 10: Pin configurations of LF15AB [5]

LF33CDT-TRY type regulator will be used to ensure 3.3V level, which has the following specifications:

- Output voltage: 3.234V-3.336V
- Operating input voltage: 16V(max)
- Output noise voltage: 50uV
- Control input logic low: 0.8V
- Control input logic high: 2V
- Temperature range: -40^{o}C – 125^{o}C

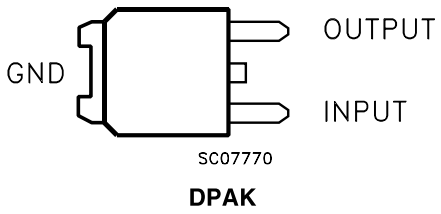


Figure 11: Pin configurations of LFCDT-TRY [5]

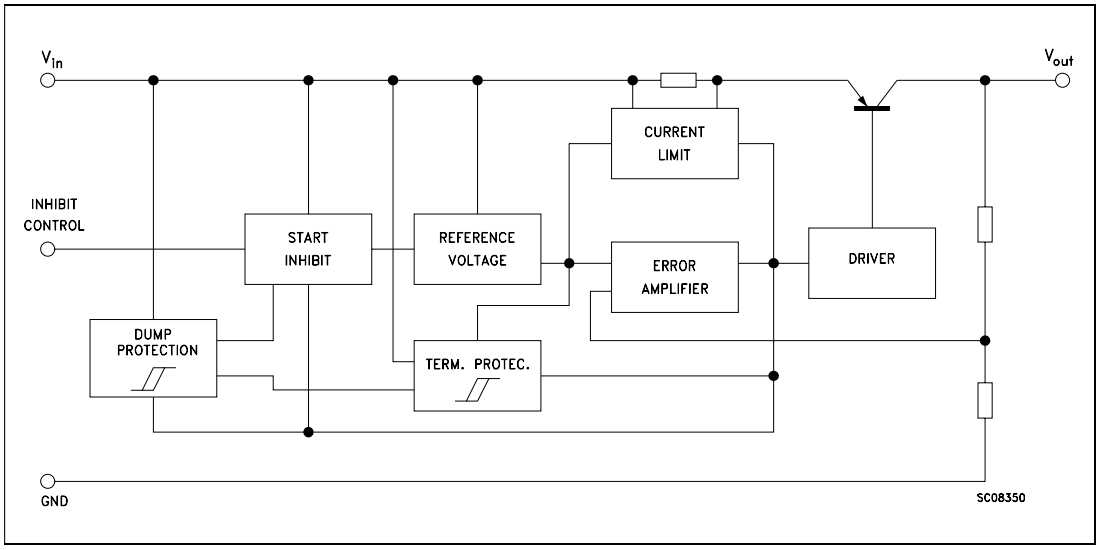


Figure 12: Block Diagram for LFxx type voltage regulators

4. APPENDIX - Key Generation Algorithm Research

RSA Key Encryption Algorithm is given below [3]:

- Generate two large random primes, p and q, of approximately equal size such that their product n = pq is of the required bit length, e.g. 1024 bits.
- Compute n = pq and (phi) φ = (p-1)(q-1).
- Choose an integer e, 1 < e < phi, such that gcd(e, phi) = 1.
- Compute the secret exponent d, 1 < d < phi, such that ed ≡ 1 (mod phi).
- The public key is (n, e) and the private key (d, p, q). Keep all the values d, p, q and phi secret. [We prefer sometimes to write the private key as (n, d) because you need the value of n when using d.]
- n is known as the modulus.
- e is known as the public exponent or encryption exponent or just the exponent.
- d is known as the secret exponent or decryption exponent.

First Step: Generation of two large random primes p and q.

In our project, the key size is predetermined as 128 bit. Therefore, p and q must be 64 bit length prime numbers.

Algorithm:

1. Generate a random number of length b/2. Set the lowest bit n to 1 (in order to ensure number is odd).
2. Check if the number is prime.

Most common way of testing whether a number is prime or not is Robin – Miller Test:

A simple example taken from Wikipedia [6]:

Core of the algorithm – determine if n = 221 is prime:
*n – 1 = 220 = (2^2)*55 (n – 1 = (2^s)*d)*
choose a random integer (a) smaller than n: let it be 174 in this case;
47 = 174^55 mod 221 (denote the result as x)
220 = 174^110 mod 221
Notice that 220 = -1 mod n, either 221 is prime or 174 is a strong liar. In other words, 221 is probably prime.
Choose another random integer (a = 137):
137^55 mod 221 = 188 which is equal to neither 1 nor n-1 = 220
137^110 mod 221 = 205 which is again neither 1 nor n-1 = 220
Then result shows that 137 is a witness that shows 221 is not a prime.

Explanation taken from Marker's Math Notes [7]:

Lemma: p is an odd prime. p – 1 = 2^km where m is odd. Let 1 <= a < p. Either, a^m = 1 mod(p) or one of a^m, a^2m, a^4m ... a^(2^(k-1))m is congruent to -1 mod p.

Algorithm Taken from Wikipedia webpage[6]:

Input: n > 3, an odd integer to be tested for primality;
Input: k, a parameter that determines the accuracy of the test
Output: composite if n is composite, otherwise probably prime
write n − 1 as 2s·d with d odd by factoring powers of 2 from n − 1
LOOP: repeat k times:
pick a random integer a in the range [2, n − 2]
x ← a^d mod n
if x = 1 or x = n − 1 then do next LOOP
for r = 1 .. s − 1
x ← x^2 mod n
if x = 1 then return composite
if x = n − 1 then do next LOOP
return composite
return probably prime

As it can be understood from the explanation given above, satisfaction of the condition for a randomly chosen number a does not prove that p is prime. Error probability is 0.25. Therefore, if the algorithm is run for k different randomly chosen a; error probability decreases to 0.25^k. Choosing k = 5 means error probability which is equal to 0.0009765625.

e will be chosen as a Fermat Prime (3, 5, 17, 257, 65537 – source Wikipedia). Note that e and (p-1)& (q-1) are coprime numbers. In the generation process of p and q, this should also be taken into consideration.

Second Step: Computing n and phi [3]

The original definition of RSA uses the Euler totient function φ(n) = (p-1)(q-1). More recent standards use the Charmichael function λ(n) = lcm(p-1, q-1) instead. λ(n) is smaller than φ(n) and divides it. The value of d' computed by d' = e-1 mod λ(n) is usually different from that derived by

d = e-1 mod φ(n), but the end result is the same. Both d and d' will decrypt a message me mod n and both will give the same signature value s = md mod n = md' mod n. To compute λ(n), use the relation

λ(n) = (p-1)(q-1) / gcd(p-1, q-1).

Third Step: Choosing e [3]

In practice, common choices for e are 3, 17 and 65537 (2^16+1) which are Fermat primes.They are chosen because they make the modular exponentiation operation faster. Also, having chosen e, it is simpler to test whether gcd(e, p-1)=1 and gcd(e, q-1)=1 while generating and testing the primes in step 1. Values of p or q that fail this test can be rejected there and then. (Even better: if e is prime and greater than 2 then you can do the less-expensive test (p mod e)!=1 instead of gcd(p-1,e)==1.)

Fourth Step: Computing d:

Computation of modular inverse is a deeply researched topic. Most common method is Montgomery Inversion method. Two examples of sources are given below:

A Scalable Architecture for Modular Multiplication Based on Montgomery's Algorithm

Alexandre F. Tenca, Member, IEEE, and

Cetin K. Koc, Senior Member, IEEE

IEEE TRANSACTIONS ON COMPUTERS, VOL. 52, NO. 9, SEPTEMBER 2003

http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1228516

Scalable and Efficient FPGA Implementation of Montgomery Inversion

Ertugrul Murat, Suleyman Kardas and Erkay Savas

2011 Workshop on Lightweight Security & Privacy: Devices, Protocols, and Applications

http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5749560

5.REFERENCES

[1] 2.4-GHz *Bluetooth*® low energy System-on-Chip Datasheet:

http://www.ti.com/lit/ds/swrs084c/swrs084c.pdf

[2] http://datasheet.octopart.com/EP1C3T100C8N-Altera-datasheet-2871.pdf

[3] "*RSA Algorithm" http://www.di-mgt.com.au/rsa_alg.html* Last Accessed on 09.12.11

[4] http://www.all-battery.com/SANYOLi-

ionPrismatic3.7V2000mAhRechargeableBatterywithFuse-30043.aspx

[5]http://www.st.com/internet/com/TECHNICAL_RESOURCES/TECHNICAL_LITERATURE/DATASHEET/CD00000546.pdf

[6] *"Rabin – Miller Primality Test"* Last Accessed on 09.12.2011
http://en.wikipedia.org/wiki/Miller-Rabin_primality_test

[7] *"Marker's Math Notes"* Last Accessed on 09.12.2011
http://homepages.math.uic.edu/~marker/math435/rm.pdf