OTA Updates for an IOT Security Device

Aneesh Malhotra, Ryan Thomas, Sohail Iqbar, Gerson C Dalton, Mohammad Nur

February 21, 2018

Introduction

- FPGA's can be used to enhance security in wireless sensor networks by implementing a cryptographic exchange between wireless nodes in a network.
- Advancements in network attacks, however, could potentially leave these systems vulnerable.
- In the event the system is compromised it may be necessary to implement a firmware update.
- Over-the-air updates are those that occur wirelessly, and allow the system to be updated easily without disassembly.

Functionality

- Our plan is to improve upon an FPGA enhanced sensor network, allowing for OTA -update capability.
- Updates could be used to update security or provide additional functionality to the system.
- These firmware updates must include the microcontroller and FPGA in each sensor node.
- This additional functionality must be able to implemented securely without creating potential vulnerabilities.

Top Level Design

1. The update will be located in the cloud (Dropbox).
2. The app will include an update button that instructs the system to push the update.
3. XBee will push the update the update to the MSP, which will calculate the size of the update and break it apart into components while pushing it to a memory pool.
4. The MSP then takes update data from memory and pushes it to the FPGA.

Top Level Design

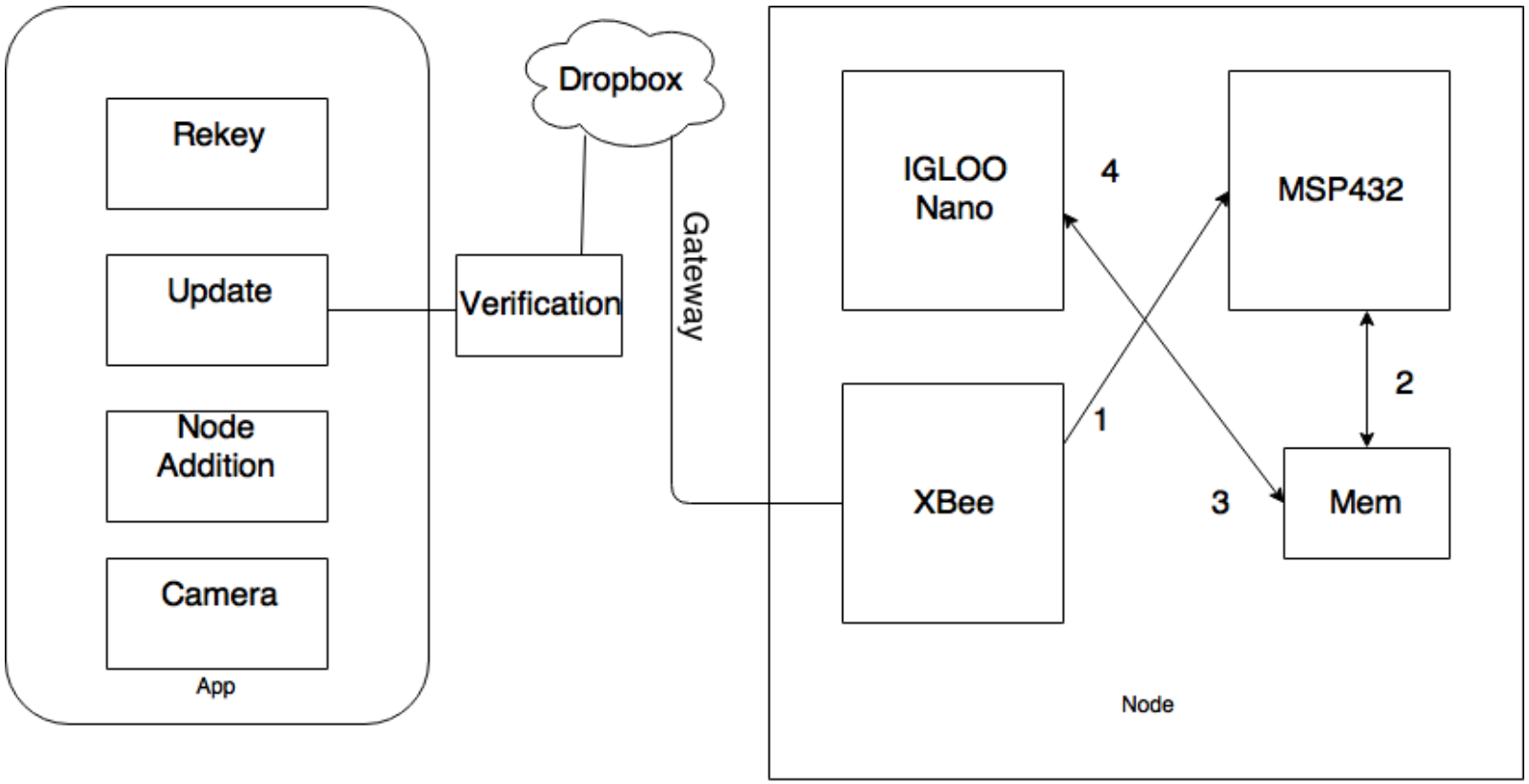
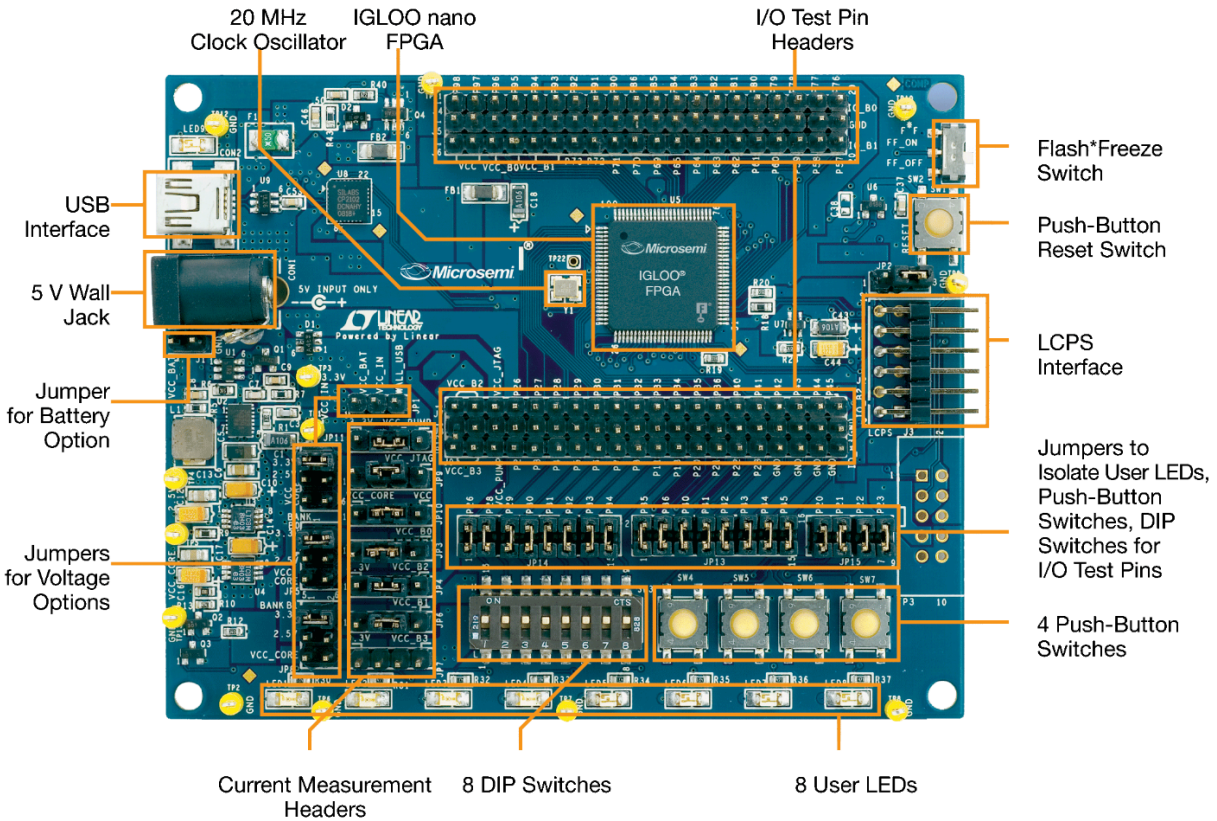


Figure: Top Level Design of OTA Update system

FPGA


20 MHz Clock Oscillator
IGLOO nano FPGA
I/O Test Pin Headers
USB Interface
5 V Wall Jack
Jumper for Battery Option
Jumpers for Voltage Options
Flash*Freeze Switch
Push-Button Reset Switch
LCPS Interface
Jumpers to Isolate User LEDs, Push-Button Switches, DIP Switches for I/O Test Pins
4 Push-Button Switches
Current Measurement Headers
8 DIP Switches
8 User LEDs
Microsemi
IGLOO® FPGA
5V INPUT ONLY
Powered by Linear

FPGA

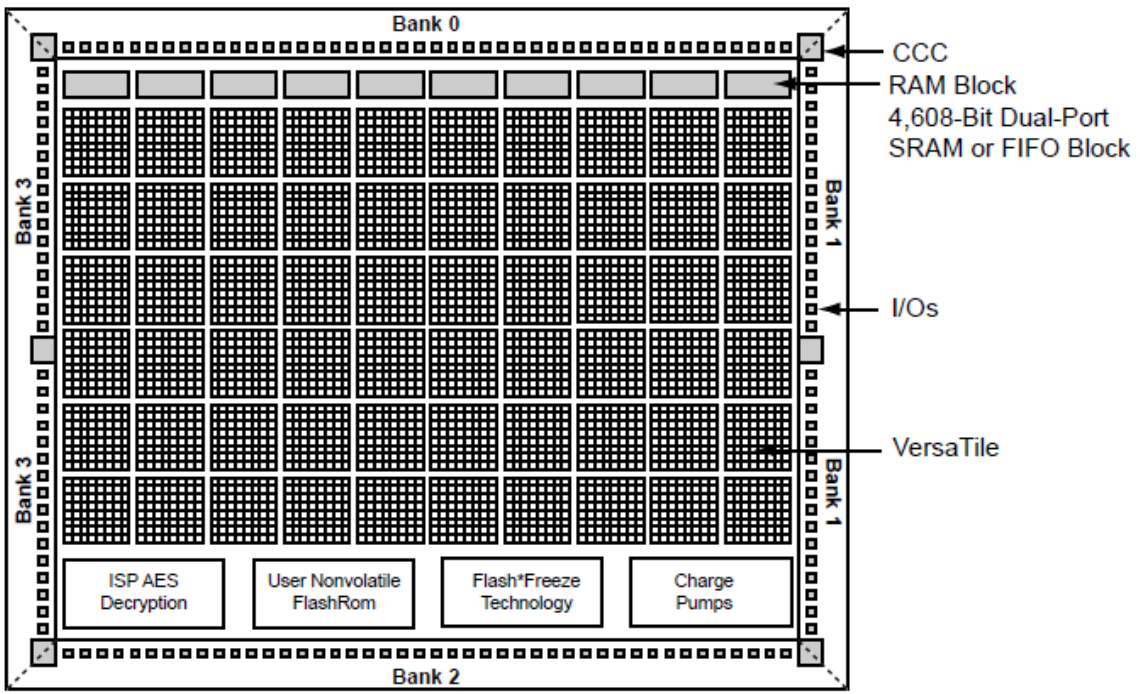


Figure 1-4 • **IGLOO Device Architecture Overview with Four I/O Banks (AGLN250)**

FPGA

- The IGLOO nano contains 1kbit of *nonvolatile* Flash ROM memory.
- ROM is written using IGLOO nano IEEE 1532 JTAG programming interface. It's content can be read back using the JTAG programming interface or via direct FPGA core addressing.
- The core can be individually programmed (erased and written), and on-chip AES decryption can be used selectively to securely load data over public networks with the security key stored in Flash ROM.
- IGLOO nano devices have embedded SRAM blocks which are 4608bits in size, and have independent read/write ports whose bit widths can be configured.

FPGA Specifications

- Clock Conditioning Circuit (CCC) and PLL
  1. Up to six CCC blocks, one with an integrated PLL
  2. Configureable Phase Shift, Multiply/Divide, Delay
  3. Wide input frequency range (1.5MHz – 250MHz)
- Embedded Memory
  1. 1kbit of Flash ROM non-volatile memory
  2. SRAM's and FIFO's with Variable-Aspect-Ratio.
  3. 4608 bit RAM
  4. Blocks (x1,x2,x4,x9,x18) organizations
  5. True Dual-Port SRAM (except x18 organization)

Data Protocols

- Our wireless sensor node will use the MSP432, which is a low power microcontroller for processing and redirecting sensor data.
- The FPGA can be made handle the same data protocols as the MSP432 such as I^2C, SPI, and UART.
- I^2C and UART protocols require 2 pins, while SPI requires 4.
- I^2C and SPI are synchronous transfers, whereas UART is asynchronous and does not require a clock.
- SPI can transfer at 16Mbps
- UART can transfer at 960kBps

Zigbee

- A high level communication protocol based in IEEE 802.15.4, which is used in low rate wireless personal networks.
- Zigbee is common in IoT devices as an alternative to WiFi.

Pros of Zigbee

- It is designed for small devices for low power consumption.
- It is a mesh network standard which can use other devices to pass signals over long distances.
- The data rates vary from 20kbps – 250kbps in the 2.4GHz band.
- Has good performance in environments with low SNR.
- Secures data through 128-bit symmetric encryption keys.
- Needs less than 64kb of ROM and 2 – 32kb of RAM.

Encryption/Security

- OTA update capability can provide a potential vulnerability to the system.
- We need a system of authenticating updates as to prevent the system using a false update.
- We can do this using a digital signature or cryptographic hash function.