# High Level Design

Team TrakPak
Bernard Floeder, Oliver Hanes, Michael Lindt & David Surine

## Table of Contents

## 1 Introduction

This project involves implementation of devices common in today's world in order to help people stay organized and confident that they have what they need for the day. By incorporating RFID tagging capabilities, as well as GPS tracking, we hope to create a backpack that will assist people by letting them know what is inside their bag, when they last interacted with each item in their bag, and where their bag is at all times. The project follows the mentality of the Internet of Things by allowing people to interact with their belongings via WiFi through an interfacing smartphone app.

## 2 Problem Statement and Proposed Solution

Losing track of one's things is a common problem for everyone. For students, belongings such as laptops, notebooks, calculators and water bottles are used on a daily basis, but they are incredibly easy to misplace. This is a common problem for all students, and the TrakPak is our solution to this problem. The TrakPak will feature an RFID reader that students can use to scan their belongings every time that they place them in or remove them from the backpack. They only need to add an included passive RFID tag to their item, and from that point on, they will be able to scan the item. When an item is scanned, the TrakPak system will note which item was scanned, the location via a GPS chip, and the time. That information will be stored and then sent to an app via WiFi that the user will be able to access. From the app, the user will be able to view a running log of when and where their water bottle, laptop, notebook, etc. was scanned on their backpack. This system should assist any student in keeping track of their belongings as well as being mindful of each time that they remove it from their backpack. Finally, using the same GPS chip, the backpack will be able to send its location to the user app. This is important because losing a backpack can be just as easy as losing a water bottle for a busy student.

## 3 System Requirements

### 3.1 Embedded System Requirements

- Microcontroller in the PIC family
    - Choose from PIC family because of familiarity with components/ programming

        - All required passive components (Vcap, etc.)
        - Programming Requirements
            - Language
                - C
            - Environment
                - MPLAB IDE
            - Hardware
                - PICkit 3
- General Requirements
    - Must be able to communicate with all auxiliary hardware, including but not limited to:
        - GPS chip
        - Clock Hardware
        - LED Hardware
        - WiFi chip
        - RFID Scanner
    - Must have enough available interfaces for all devices (SPI, I2C, general I/O pins, USB, etc.)
    - Each auxiliary hardware piece must have an available communication protocol (SPI, I2C, UART, etc.)
    - Must have Real Time Clock hardware and protocol available
        - Ex: PIC32695 has RTC available

### 3.2 Power Requirements

- General Requirements
    - Input: Must be rechargeable via USB connection or connection to wall
    - Output: Must be able to connect to our microcontroller via a microUSB connection
    - Weight Requirement
        - As light as possible considering that the user will have to carry it around
        - Maximum: 5 lbs
    - Size Requirement
        - Shape of battery must not be bulky such that integration into the backpack is difficult and/or the backpack is not as usable
        - Actual dimensions are hard to specify since possible batteries come in all shapes and sizes
    - Total Maximum Power Draw: ~500 mA

- The above power estimate has been made by choosing a likely candidate for each component and then determining its maximum power draw. This should give us a worst-case scenario power estimate.
    - Microcontroller
        - PIC Datasheet
            - Maximum Input Current = 300 mA
    - GPS chip
        - [illegible]
            - 30 uA (standby current) at 3.3V = 54 uW
    - LED Hardware
        - Average Diode Forward Current = 150 mA
            - Depends on current limiting resistor choice
    - WiFi chip
        - ESP8266

        - 15 mA typical sleep value, 170 mA TX max, 50 mA RX max
    - RFID Scanner
        - ID-12 LA Datasheet
        - 35 mA maximum
    - Required Runtime: 12 hours
    - Required Battery Estimate: 8556 mAh

### 3.3 Wireless Interface Requirements

- General Requirements
    - Must be able to communicate with WiFi chip connected to the central PIC board and RFID scanning device
        - Must be able to connect to known, private networks and connect to any unsecured, public networks
    - Phone system and WiFi chip both must be able to communicate within the domain of the created mobile app

### 3.4 User Interface Requirements

- General Requirements
    - Must be able to alert the User that each item has been successfully or unsuccessfully scanned
        - If there is an error in scanning in an RFID tag the LED will flash red
        - If an RFID scan is completed the LED will flash green
    - Must be able to display the remaining life in the battery or alert the user when the battery is getting low
        - Can use Coulomb counting capabilities to track current and display that to the mobile app
    - Must have a functioning mobile app to provide scanned information for each item (Object, GPS & Time Stamp)
        - Ideas on how to implement forthcoming; more knowledge required for necessities for building a mobile phone application

### 3.5 Installation/Use Requirements

- General Requirements
    - The TrakPak will not be user-installed; we will install and assemble all components.
    - The passive RFID tags will be installed by the user onto belongings of their choice via stickers.
    - The following components should be able to be installed directly on or connect directly to a PCB of dimensions ~4 in x ~3 in.
        - Microcontroller (and all its required passive components)
        - Clock Circuitry
        - GPS Chip
        - WiFi Chip
    - The RFID scanner must be installed in an easy-to-reach area of the interior of the backpack to protect it from weather.
    - The User LEDs will be installed on an easy-to-see area of the outside of the backpack near the RFID scanner. This way, a successful scan can easily be verified by the user.
    - The battery and microchip must be installed in a convenient location within the backpack to prevent damage to the battery and preserve interior space.
        - The battery's power cord must have a way to be secured when not in use.
    - All wiring must be secured and well hidden to prevent damage from wear and tear as well as preserve a sleek look.

### 3.6 Safety Requirements

- Must ensure that the battery can account for overcharging without overheating too much.
- Ensure that the battery charging process is well protected to avoid delivering shocks to the user.
    - Only possible dangerously high voltage is from charging the backpack off the wall.
- As a backpack, do not overburden yourself to avoid damage to spinal system

### 3.7 Mechanical Requirements

Our project is designed to improve a user's experience with a backpack. As such, there are several main mechanical requirements that cannot be exceeded or the

TrakPak would cease to be an advantageous product. Primarily, the TrakPak must be able to be fully functional inside of a backpack without compromising the carrying capabilities of a comparatively sized backpack. In order to meet this expectation, we will have to use hardware that is small, lightweight, and durable.

The first consideration is size. Our hardware will need to be small enough to be incorporated into an existing backpack without taking up too much space. The TrakPak would be useless if there was no room to put anything into it. Therefore, we will need to buy parts that will minimize space while keeping effectiveness high. The major challenge for this consideration will be the power supply, since most power supplies are relatively large compared to the average microcontroller. The specific size requirement will be determined by the backpack that we choose to augment, however as a general rule we will need to buy the smallest parts possible that fit our requirements.

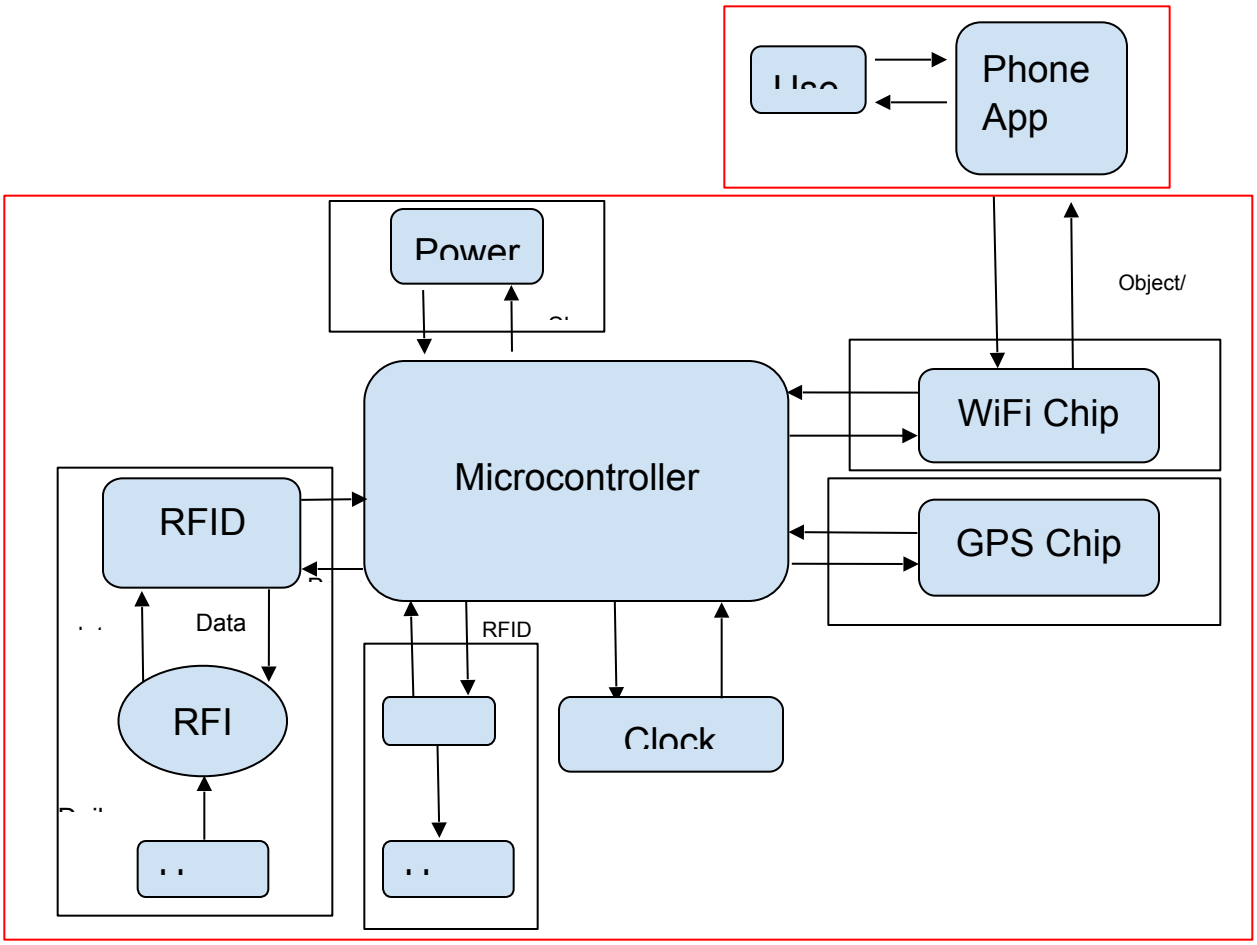
The second consideration is weight. Backpacks quickly become uncomfortable when they are filled with too much weight. While we may not be able to decrease how much a user tends to put inside the TrakPak, we can control how much the base weight is for an empty TrakPak. In order to keep this base TrakPak weight at a reasonable level, we will need to find parts that have minimal weight. Most of our components, like the microcontroller or the LEDs, will be of negligible weight. However, the power supply may be another challenge as batteries with the power capacity that we will need tend to be comparatively heavy. The specific base weight requirement for the TrakPak will be highly dependent on the weight of the backpack we choose to augment, however our other components should be under five pounds altogether.

The third consideration is durability. The TrakPak would be a highly ineffective product if it couldn't stand up to the rigors of daily life. Few backpack users treat their backpacks with special care. They are often dropped on the ground, stepped on, or in other ways misused. We cannot assume that the TrakPak will be treated any differently than a regular backpack by a user, and so our components will need to be able to survive these daily abuses. There are two ways we can meet the durability requirements. The first is by purchasing already durable parts. Some parts, like the microcontroller and its immediate plugins will be inherently fragile, but other parts, like the power supply or the RFID scanner, may have variable durability. The second way to meet our durability requirement is by packaging our parts. This packaging will need to meet the size and weight considerations, as well as sufficiently protect our hardware from the daily wear and tear of backpack usage.

In conclusion, the TrakPak is dependent on three mechanical requirements. Firstly, the hardware must be small enough to not hinder the TrakPak's carrying capacity. Secondly, the hardware must be light enough to not inhibit the user's comfort. Thirdly, the hardware must be protected so that the TrakPak's technological components remain functional. These three mechanical requirements will ensure the

longevity of the TrakPak functionality and the user's satisfaction with the physical construction of the TrakPak.

## 4 System Block Diagram




### 4.1 Overall System

The following outline shows our overall system and its subsystems. The overall system and its subsystem will be subject to the requirements that we outlined in Section 3. We list each section's purpose and required inputs and outputs in 4.1 and 4.2, and give our proposed solutions to fulfill that purpose in Section 6. This outline matches the figure shown above.

- Overall System - TrakPak System
    - Purpose - A backpack that assists the user keep track of their belongings
    - Inputs:
        - User scanning of a belonging
        - User request for data regarding status of belongings
        - Power
        - GPS Information
    - Outputs:
        - Data regarding status of tracked belongings
        - Feedback on a successful scan of a belonging
    - Subsystems
        - Phone App
        - TrakPak
            - Subsystems
                - Microcontroller
                - LED Interface
                - GPS Module
                - Wifi Module
                - RFID Module

### 4.2 Subsystem and Interface Requirements

The following outline gives each subsystem's general requirements. There are also requirements from Section 3 that these subsystems are subject to if they apply. Again, we are listing all the required inputs and outputs for each of these modules.

- Subsystems
    - Phone App
        - Purpose: An interface between the user and the Trakpak
        - Inputs:
            - User data request (via touchscreen)
            - Data from Trakpak
                - GPS Information
                - Real Time Information
                - Item Identification
        - Outputs:
            - Data from Trakpak
                - GPS Information
                - Real Time Information
                - Item Identification
            - User data request

- TrakPak
    - Subsystems
        - Microcontroller
            - Purpose: To coordinate all signals and power in the Trakpak
            - Inputs:
                - Vdd from power
                - Data from RFID Scanner
                - Data Request from LED
                - GPS data from GPS chip
                - Data request from WiFi chip
                - Clk from clock
            - Outputs:
                - Sleep state to power
                - Power On/Off to RFID Scanner
                - RFID read complete to LED
                - Data request to GPS chip
                - Voltage to clock
                - Data to WiFi chip
        - LED Interface
            - Purpose: To visually alert the user to a successful scan
            - Inputs:
                - RFID read complete
            - Outputs:
                - Light on to show user scan is complete
                - Data request to Microcontroller
        - GPS Module
            - Purpose: To acquire GPS data upon request
            - Inputs:
                - Data request from Microcontroller
            - Outputs:
                - GPS data to GPS Chip
        - Wifi Module
            - Purpose: To communicate between the User App and the Trakpak via Wifi
            - Inputs:
                - Data from Microcontroller
                - Data request from phone app
            - Outputs

        - Data request from Microcontroller
        - Object/GPS/Time Stamp to Phone App
- RFID Module
    - Purpose: To allow for the scanning of items
    - Inputs:
        - Power On/Off from Microcontroller
        - Data from RFID tags
    - Outputs:
        - Data to Microcontroller
        - Data request to RFID tags

### 4.3 Future Enhancement Requirements

Though the initial release of the TrakPak will have many appealing features, there are many that we had to leave out for the initial release.The first feature that will be included on future releases is an RFID reading feature that doesn't require scanning items into and out of the backpack everytime it happens. We will implement an RFID scanner that will iterate every 5-10 mins and tell the user every item inside the backpack. Another feature that we will include on future iterations is a feature to tell the user where and when an item was taken out or put into the backpack. This will improve the user's ability to track down lost items.

## 5 WiFi considerations

### 5.1 Connection Considerations

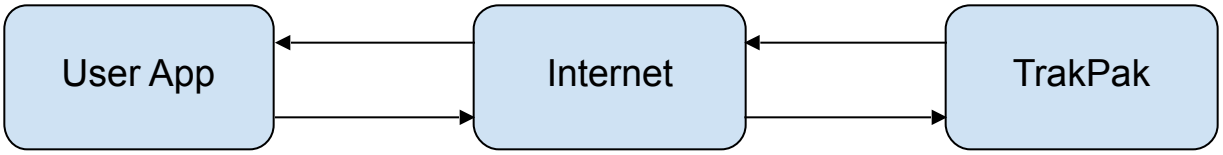
This device will be connecting to multiple networks due to the fact that someone traveling with this backpack will be going to multiple locations where WiFi is present. The connection should be able to connect to all unsecured networks and all known private networks (such as home or work WiFi). The setup of the connection will be managed by prioritizing connecting to unsecured networks if they are in range. If there are no unsecured networks present then it will try to connect to any of the known secured networks. If these are unavailable the WiFi chip should be searching for available networks every few minutes.

### 5.2 Data Flow

Data flowing from the TrakPak to the User App will consist of a 32 bit identifying number read from the RFID tag, a GPS latitude and longitude number and a real time number. Data flowing from the User App to the TrakPak will consist of a small bit number that simply requests GPS information from the Trakpak.

At any one time, we are only sending three or four numbers at most, so we will not be sending much information at all. We would be sending data only as often as items are being scanned, or as often as the user manually requests the TrakPak's GPS location.

Some kind of server will be involved, but at this point, we do not have the specifications for that. We likely will try to set up a server on the User App side, so that we can store and present the data in a user-friendly way. However, we do not know how to do that yet.




### 5.3 ESP8266

To use the ESP8266 we will be processing information using the PIC microcontroller and then sending the information to the ESP8266 to be transmitted via WiFi. The model that makes sense for our project is one in a kit that will connect to our board externally, for example this model. We plan to program this chip using SPI communication because it will send data at a rate fast enough for the project and it is the programming environment that the engineers in our group feel more comfortable with. Along with those reasons, we have SPI capabilities on our board that otherwise are not being used so it would be efficient to use it for programming our ESP8266.

### 5.4 Protocol for Data Exchange

We are planning to use IEEE 802.11 protocol for WiFi because it's pretty standard and should be relatively easy to implement. Since we do not know how the app design and implementation is going to look, we cannot say for certain how the rest of our communication protocols will be setup. That will have to be determined early on next semester.

## 6 High Level Design Decisions

- Phone App
  - Purpose: An interface between the user and the Trakpak
  - Inputs:
    - User data request (via touchscreen)
    - Data from Trakpak
      - GPS Information
      - Real Time Information

        - Item Identification
    - Outputs:
        - Data from Trakpak
            - GPS Information
            - Real Time Information
            - Item Identification
        - User data request

To implement the phone app requirements, we are planning to write an iOS application. iOS applications allow for easy user inputs via touchscreen. They allow for a nice display of information, which we will need to show Trakpak data to the user. There is also available software and lots of documentation out there for developing apps, which is good because no one in the group has significant developing experience.

- Microcontroller
    - Purpose: To coordinate all signals and power in the Trakpak
    - Inputs:
        - Vdd from power
        - Data from RFID Scanner
        - Data Request from LED
        - GPS data from GPS chip
        - Data request from WiFi chip
        - Clk from clock
    - Outputs:
        - Sleep state to power
        - Power On/Off to RFID Scanner
        - RFID read complete to LED
        - Data request to GPS chip
        - Voltage to clock
        - Data to WiFi chip

We are currently looking to use a microcontroller from the PIC family in our project implementation because it is what we have worked with before, which would make programming it easier than choosing a new microcontroller we have not worked with. From our understanding, there are plenty of PICs available that will satisfy our requirements. One thing that we need to weigh is whether it is easier to run our system using MPLAB X IDE, or if in order to easily use the WiFi module we need to use an Arduino.

- LED Interface

        - Purpose: To visually alert the user to a successful scan
        - Inputs:
            - RFID read complete
        - Outputs:
            - Light on to show user scan is complete
            - Data request to Microcontroller

To implement the LED interface, we are planning to use a green LED that is secured to a small strip of breadboard. That breadboard will be easier to secure within the strap of the backpack to ensure that the LED is in a highly visible area. The breadboard will also make it easy to connect the LED to the microcontroller via some wiring through the backpack. This setup should satisfy all LED module requirements.

- GPS Module
    - Purpose: To acquire GPS data upon request
    - Inputs:
        - Data request from Microcontroller
    - Outputs:
        - GPS data to GPS Chip

To implement the GPS chip, we are planning to attach the chip directly to our PCB, and it will be able to communicate its location coordinates to the microcontroller. Therefore it will satisfy our requirements as well as be a cost effective solution to keeping track of the coordinates of the backpack.

- Wifi Module
    - Purpose: To communicate between the User App and the Trakpak via Wifi
    - Inputs:
        - Data from Microcontroller
        - Data request from phone app
    - Outputs
        - Data request from Microcontroller
        - Object/GPS/Time Stamp to Phone App

As recommended, we plan to use the ESP2866 chip to implement the WiFi module. It should satisfy all requirements of our WiFi module, and we will have support in successfully programming this chip to communicate with both the microcontroller and the user app.

- RFID Module

- Purpose: To allow for the scanning of items
- Inputs:
  - Power On/Off from Microcontroller
  - Data from RFID tags
- Outputs:
  - Data to Microcontroller
  - Data request to RFID tags

We plan to use the ID-20LA chip from Sparkfun, its surrounding starter kit, and passive 125 kHz sticker tags to implement the RFID module. The sticker tags can be applied to any item that the user wants to keep track of. The ID-20LA should satisfy the requirement of scanning the ID tag, and the starter kit provides some pieces that can communicate to a microcontroller. Thus, these components should satisfy the requirements, and they are cheap, small and lightweight.

### 7 Open Questions

Our group is still uncertain as to how the mobile app will work as well as what kind of functionality it needs in order to receive information from our different subsystems via WiFi. We will need to determine what kind of address the app will have and how to communicate tracking data to that address.

We are not sure how to connect the TrakPak to private networks. We are also unsure of how to connect the TrakPak to public networks that require the user to accept terms and conditions before granting access.

### 8 Major Component Costs

- Backpack - $10 base price
- http://www.eastsport.com/eastsp

ort-spo
rt-
bac
kpa
ck-
32
27.
ht
ml?
gcli
d=
CI
Pzi
p_
wlc
kC
FQ
6O
aQ
od
8io
G7
A

- RFID Reader
  - SparkFun - $35 base price
    - htt
      ps:/
      /
      ww
      w.s
      par
      kfu
      n.c
      om
      /
      pro
      duc
      ts/
      118
      28
    - Reads 125 kHz
- RFID Tags
  - Parallax 125 kHz tag - $1 - $3 each, base price
    - htt

p://www.robotshop.com/en/parallax-rfid-tag-125khz.html?gclid=CMbVvMvxlckCFQeNaQod0UoCGg

- Battery
  - Have estimates for 8000mAh batteries on Amazon for around $20-$30
    - http://www.amazon.com/Jid

eTech%C2%AE-Universal-Waterproof-8000mAh-portable/dp/B017R821EU/ref=sr_1_1?ie=UTF8&qid=1449436570&sr=8-1&keywor

ds=external+USB+battery+8000mA+hour

- http://www.amazon.com/dp/B00KXKJRV0?psc=1
- PCB Board - $50 base price

## 9 Conclusions

This project is, at its core, to create a product that will help consumers stay organized and prepared for life's daily challenges. The TrakPak will allow users to ensure that they have all of the things they need for day to day success through the use of several technologies. By utilizing technologies such as Wifi, GPS tracking, RFID scanning, a microprocessor, and a mobile app, the TrakPak will be able to keep track of where users last interacted with their belongings, what items are in their TrakPak, and where to find their backpack if it goes missing. These functions empower the user to be confident

that they are ready for the challenges that fill day to day life.  Even when they aren't prepared, the TrakPak reminds the user of what he needs for the day and where to begin looking for lost items. The TrakPak is just another example of how the Internet of Things can be so influential in our lives.2

**10 References**

All things that we reference are directly cited/hyperlinked within the document sections.