Ruby Ride tracking unit timeline

What steps will I need to take to pull this off? To answer that we'll need to look at what the final product will need to do. This tracking unit will be used for vehicle access control for employees. It will need to take an employee ID card or RFID tag and send it to a server to figure out if the employee has access to the vehicle and then sends a message back to the tracking unit to lock or unlock the vehicle doors to allow access to the vehicle. Using timestamps, this technology would also be able to be used for other administrative tracking since this shows each employees driving activity when in company vehicles.

The information here is all focused on the path toward making a working prototype that can be replicated. After a prototype is finished and deemed acceptable, replication and mass installation can take place which at that point this solution will be fully implemented. Priority will be given to bringing the GPS and GPRS/GSM function to commission, so a final GPS tracking solution will need to be finished prior to moving on to employee RFID access.

- **Function:**
    - Detect GPS position and RFID tag information and transmit this to an Amazon rental server for processing. Final product must also be able to receive lock/unlock commands from the server and trigger a keyfob button to lock or unlock the vehicle doors.
    - Broken down functions
        - RFID read capability
        - GPS capability
        - 3G/4G internet capability (GSM/GPRS)
        - Microcontroller
- **Operating environment**
    - Final device will need to be either powered from the vehicle harness or have some other reasonable solution for power such that it doesn't have to be serviced too often.
        - If powered from the vehicle harness, the tracking unit will need to be fuse protected and have protective circuitry to function in the electrically noisy and high power environment.

The base of the tracker, being the microcontroller, will be found through a review of a number of controller solutions comparing cost, processing capability, and will focus more on arduino IDE compatible controllers but not necessarily arduino brand or form controllers. This choice will be made

to lower the learning curve because of the ease of use of the arduino platform. If this is how the final product can be put together, here is the breakdown of the parts needed;

- **RFID Function**
    - Arduino compatible RFID shield
    - Enough tags to simulate different vehicle access
- **GPS function**
    - Arduino compatible GPS module
- **3G function**
    - Arduino and 3G compatible GPRS/GSM module
    - Sim card compatible with 3G (with contract or pay as you go)
- **Keyfob Control**
    - No extra hardware needed. This is built into the microcontroller
- **Safety**
    - Automotive transformer or other type of current shunt for limiting the power transferred to the controller to a safe level.
- **Container**
    - Packaging for protection of the tracker circuitry and prepare for simple installation.

This means for a prototype, I'll need to acquire these add-ons making sure that they don't require the use of the same pins on the microcontroller itself, test each module individually to show it works, then combine those functions based on the individual tests.

How will I actually test this?; There are a limited number of types of all of these modules among the group that are arduino compatible. They also have spec'ed ways of interfacing with them along with a wealth of open source documentation and programming libraries. I'll gather documentation for each module and iteratively try different ways of interfacing with it based on what I can find in it's documentation until successful. Should I get stuck, there is a vibrant knowledgebase and community around the arduino platform and many other popular microcontrollers to draw from including plenty of RFID GPS and I.O.T. (Internet of things, read wireless internet access) examples.

In concrete steps, how do I get to my end goal?;

1. **Research technology and purchase materials-**

    This time will be spent finding RFID, GPS, and 3G modules that are compatible with the arduino platform and each other. When doing this I'll need to pay attention to the amount of

documentation available, and what microcontroller connections each module and their respective programming libraries use. If one module uses the same pins as another, the modules library source code will need to be modified, which introduced more complexity and introduced opportunity for error and lagging. I'll need to pay close attention to RFID reading range, GPS accuracy, and make sure that the GSM/GPRS module is actually 3G compatible, which seems to be hidden on a lot of boards. I'll need to figure out how others are protecting circuitry that needs to operate in such a dangerous environment, and still safely operate without draining the car battery. I'll need to try to address an RFID reception issue that could possibly happen since many passive tags only have a range on the order of ~2 inches. At the end of this step the price per unit for the hardware alone can be estimated.

2. **Test each module until consistent, controllable, and predictable behavior is achieved-**

   This is an iterative step that I'll need to do for each module progressively. I'll operate each module separately from each other and test changing around the code to show that the module behaves acceptably. GPS and RFID will be achievable using only the modules and a microcontroller with the addition of the RFID tags. The GSM/GPRS module will need a destination test server that I have access to for testing. Finding a source for this will take extra time.

3. **Combine module libraries and circuitry one at a time to assemble full function-**

   I'll combine simpler modules first and test to make sure they both work, then add the third module and safety circuitry.

4. **Power Analysis-**

   Here I'll do the crucial power analysis to check to see how much power the whole unit is using and calculate the maximum power that can safely run through each part of the final product. I'll measure max and min current drawn with each module connected alone and then with the unit fully assembled. Then I'll analyze how long it will take to drain an automotive vehicle. Here is where I figure out how this device will be powered and what our best options for something like sleep mode is. The trick is, if the whole device is off, the RFID module can't read a tag, and therefore can't set in motion the steps to unlock the vehicle doors.

5. **Packaging and in field testing-**

   Here, I'll try different mounting locations in the vehicle itself to check if there are any problem areas or optimal configurations. I'll also test RFID reception in different areas of the vehicle and see if the RFID tag range is acceptable. After this step, development cost and installation per vehicle price will be better estimable.