Minutes

Location: Shed
Date: Tuesday 24th October
Not In Attendance: N/A

Attendance: Daniel Knox, Daniel Carl Beauchamp, Dharius Robinson, Natalie Mclaren

What's Been Done since the Previous Meeting:
<Daniel B>: LoRaWan pseudo code
<Dharius>: Ultrasonic sensor pseudo code
<Natalie>: SD card read/write pseudo code

Topics discussed:

- **Questions on the processor:**
  - Dan K mentions the SAM D:
    - It has an RTC - used to wake the device up from its sleep.
    - Higher clock speed so can do number stuff quicker.
  - The ATmega328P downside:
    - There is only one hardware serial.
  - Daniel B asks whether there is a preference on the processor:
    - Dan K explains that if we choose a 32 bit one, he suggests we use the same one as in the Adafruit board.
  - Dan K mentions the Arduino DUO or Arduino ZERO to solve the 'one hardware serial' issue. Both of these have two USB ports - a programming port and a serial port.

- **The sensor:**
  - When speaking about lasers:
    - Dan K suggests we find information on laser distance finder.
    - Can get off the shelf ones - namely Sharps.
    - Dan K suggests we look into wall obstacle avoidance robots as they use the type of laser we would need.

- **The battery:**
  - Dharius asks whether there are any resources to calculte life expentancy based on the processor?
    - Dan K explains that most batteries will come with a datasheet, stating the curve of voltage over time. This can be used to calculate life expentancy.
    - All batteries will decreased in voltage as you use power.
    - Look at G-Labs - they offer a lot of articles on lower power/getting the power lowered, different batteries available.
    - Our main concern should be capacity over time vs current consumption more than weather/temperature conditions affecting it.

    - Dharius asks whether it is sensible to look into disposable batteries giving our use of the sensors:
        - Dan K says we should be interested in the current - can our battery tech supply that current?
        - Dan K explains we would typically get coin cells or AA batteries for larger stuff.
        - We should watch out for potential brown outs - when some of your components can run and others can't. Not enough power for everything.
        - Knowing voltage range is important - should determine a point at which we need to tell the components to stop working.
        - A lot of digital sensors will have a sleep mode nowadays. To wake them up/shut them down if needed.
        - If there is no sleep mode available - we can use switches.
        - We can also have capacitors - which act as mini battery banks. Some components have a reserved battery.
    - Dan K mentions a common battery strategy: to only report things that aren't normal giving recent readings. Like a major change.
        - Otherwise, just one message a day to confirm its working.
    - Dan K mentions the important thing is finding out the size of our data structure and whether our device can give us that.

- **Our pseudo code:**
    - Dan K explains that our pseudo code should be in plain english. But having looked over our code, he comments on the following:
        - The ultrasensor code:
            - We should be using int instead of float.
        - The SD card code:
            - We should use codes instead of strings for when it comes to reporting errors/crashes. Not as huge as a string and quicker to read.
    - Dan K suggests we implement some sort of debugging tool i.e. an engineering menu. So our device can check that it still can communicate with each component and that in turn each component is working. Good in the case it just stops working or when first set up.

What's Being Done:
Further work on our pseudo code.

To be discussed on our individual group meeting.

Further Discussion:

Video on a useful engineering menu we could use as a guideline
https://www.youtube.com/watch?v=QK9ZYrqPHfI