**Memo**

**Senior Design**

ENG EC 463 / 464

To: Professor Pisano

From: SmartLoo (Arturo Asmal, Ben Corn, Bonnie To, Brandon Ng)

Team: SmartLoo: Team 20

Date: 4/29/18

Subject: SmartLoo - Resource CD Hardware Report

1. **Overview**

The SmartLoo system is comprised of an interconnected system of SmartLoo Sensor Nodes, installed throughout targeted areas in a restroom, paired to a SmartLoo Bridge. The SmartLoo Sensor Nodes are composed of a custom designed PCB, connected to an XBee S2C radio communication module, AA battery pack, and the appropriate sensor required to track the status of the targeted area of the restroom. The PCB is connected to its sensor via the SmartLoo Wiring Harness that is slotted into the six pin header embedded into the side of the enclosure. Also embedded into the enclosure is a slide switch, able to turn the SmartLoo Sensor Nodes on and off.

The SmartLoo Sensor Nodes transmit the data collected, from their accompanying sensors, to the SmartLoo Bridge via XBee radio, serial communication. Once the SmartLoo Bridge has received a message from a SmartLoo Sensor Node, the SmartLoo Bridge parses the message and forms a POST request to the SmartLoo Web API.

**2.0 SmartLoo Sensor Node**

2.1 **PCB Design**

2.1.1 **EAGLE**  
The SmartLoo Sensor Node PCBs were designed in Autodesk EAGLE. All of the final schematic, layout and gerber files can be found in the SmartLoo/HW/Resources/EAGLE directory of the Resource CD. There are a total of three revisions of the SmartLoo Sensor Node PCBs, with LooSensorNodeV2 being the most recent revision.

2.1.2 **SmartLoo Sensor Node V0**

The SmartLoo Sensor Node V0 PCB was designed to prove the feasibility of designing custom hardware capable of replacing the commercially purchased hardware used for the First and Second Deliverable Testing. The board design was based on the Arduino Pro Mini 3.3V board and included minor defects that were corrected in the SmartLoo Sensor Node V1 PCB design. The design also broke out all of the digital and analog I/O pins connected to the ATmega328P-AU.

2.1.2.1 **Power Supply**

The V0 design included a two pin JST header to connect to Adafruit Li-Po 2S battery pack. Connected to the power line are 3.3V and 5V LDO voltage regulators, which powered the ATmega328P-AU microprocessor and accompanying sensor respectively.

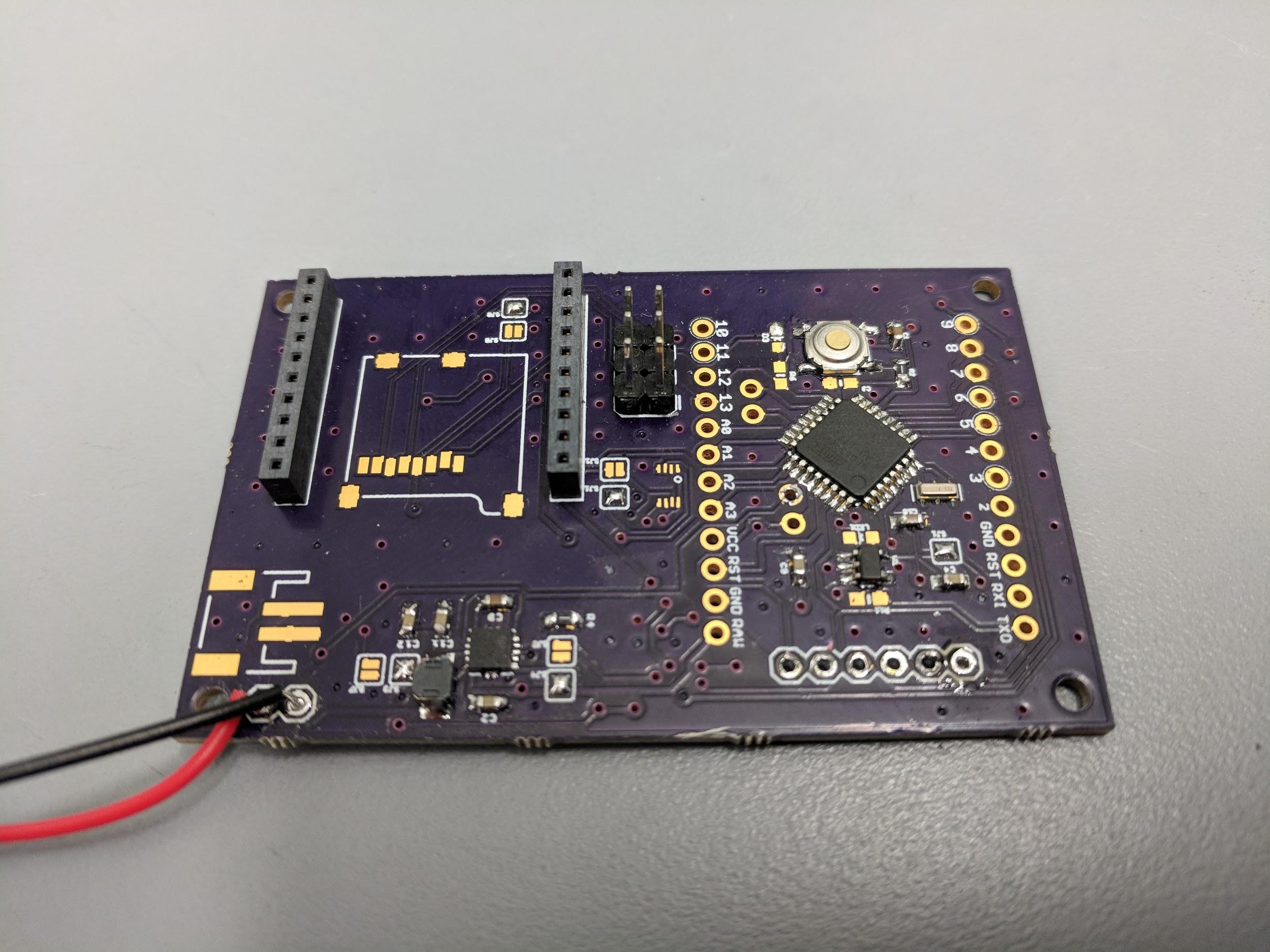
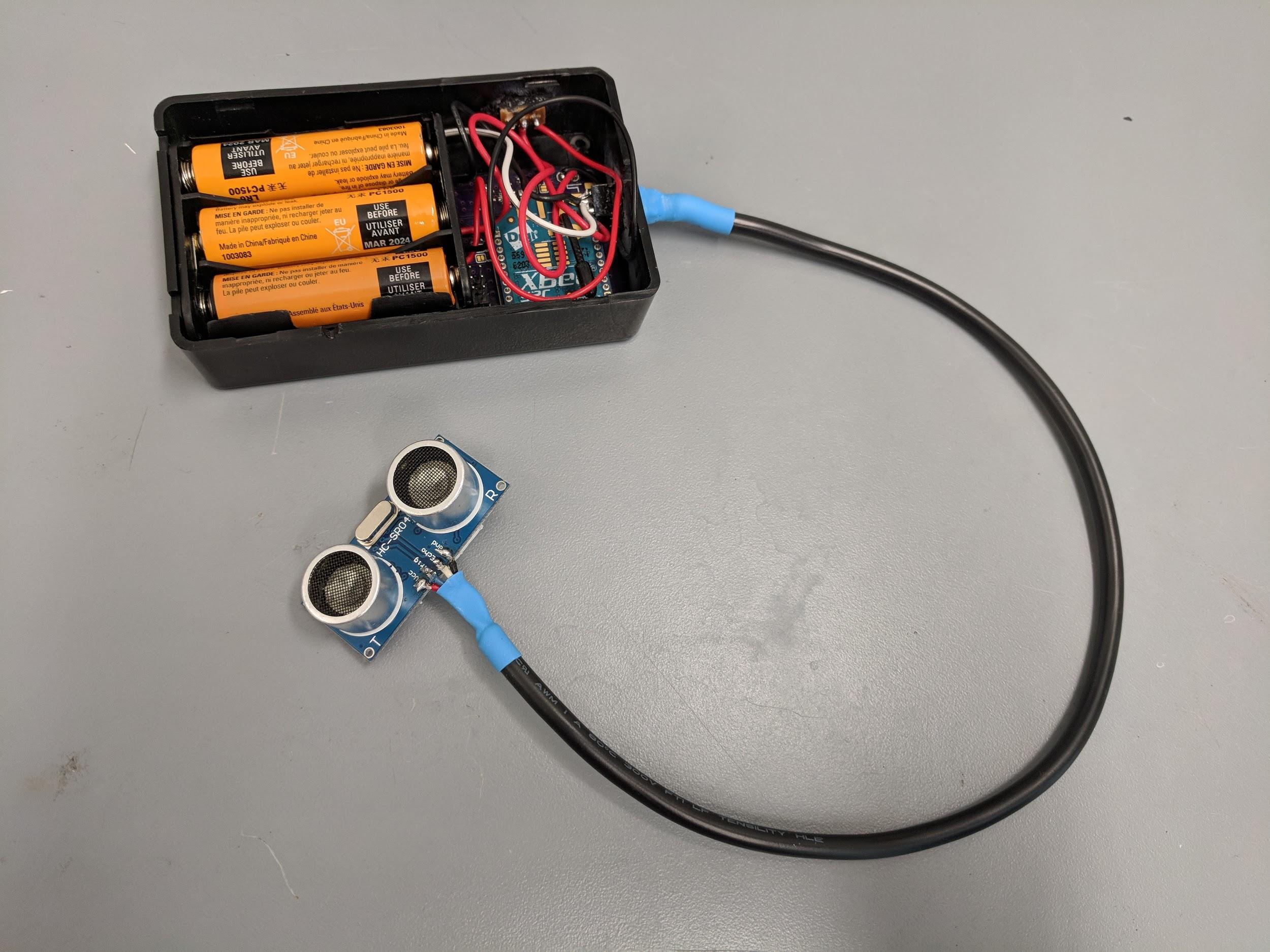
2.1.2.2 **XBee S2C**

The V0 design included a female connection interface for an XBee S2C module, so that an XBee S2C could easily be slotted into the board. Unfortunately, the pins were improperly mapped for the V0 design.

2.1.2.3 **SD Card**

The V0 design included a section to connect a microSD card holder for data storage purposes.

2.1.3 **SmartLoo Sensor Node V1**

The SmartLoo Sensor Node V1 PCB shown above, was designed to correct the defects discovered in the V0 design and offer more flexibility in the choice of the power supply.

2.1.3.1 **Power Supply**

In addition to the 3.3V and 5V LDO voltage regulators, the V1 design included a TPS61202 5V switching voltage regulator and the ability to select the power lines to be used, via solder jumpers.

2.1.3.2 **XBee S2C**

The V1 design corrected the pin connections from the ATmega328P-AU to the XBee S2C.

2.1.4 **SmartLoo Sensor Node V2**

The SmartLoo Sensor Node V2 PCB was designed to significantly reduce the size of the PCB to 1.1”x1.05”, by removing all superfluous connections and components. The V2 design is the current design of the SmartLoo Sensor Node PCBs.

2.1.4.1 **Power Supply**

The 5V LDO voltage regulator was removed from the V2 design.

2.1.4.2 **SD Card**

The microSD card holder was removed from the V2 design.

2.1.5 **OSH Park**

For the sake of quality and cost, OSH Park was selected as the PCB manufacturer of choice. OSH Park is known for their purple silk screen finish, maintains certain desirable certifications, and charges based on the square inch size of the PCB.

2.1.6 **Bill of Materials**

The SmartLoo BOM can be found in the SmartLoo/HW/Resources directory of the Resource CD. The BOM lists all of the materials required to create a Sensor Node less the PCB, which must be ordered separately. The components are sourced from Mouser, Digikey and Amazon.

* 1. **Power Supply**
     1. **Battery Pack**

The SmartLoo Sensor Node PCBs are designed to be attached to 3xAA battery holder that supplies power to the PCB, XBee S2C and sensor. The battery holder is also wired through the slide switch embedded into the side of the enclosure.

* + 1. **MIC5205**  
       The MIC5205 is a 3.3V LDO voltage regulator that takes the voltage from the batteries and steps the voltage down to 3.3V. The 3.3V power line is then used to power the ATmega328P-AU and the XBee S2C. The MIC5205 circuitry is designed to be on as long as the power switch in the on state and the batteries produce a voltage output greater than 3.3V.
    2. **TPS601202**  
       The TPS61202 is a 5V switching voltage regulator that takes the voltage from the batteries and boosts or bucks the voltage to 5V. The 5V power line is then used to power the accompanying sensor. The enable pin of the TPS61202 is tied to a digital I/O pin of the ATmega328P-AU.
  1. **ATmega328P-AU**  
     The ATmega328P-AU is an Atmel chip that has the same capabilities as the microprocessor on the Arduino Uno. When burned with the proper bootloader, the ATmega328P-AU can be used and programmed like an Arduino board via the Arduino IDE. The ATmega328P-AU was chosen for its low cost, compact form factor, ability to operate at both 3.3V or 5V, and computational abilities.
     1. **AVR-ISP MKII Programmer**

An AVR-ISP MKII Programmer in combination with the Arduino IDE was used in order to burn the bootloader onto and program the microprocessor.

* + - 1. **Bootloader**  
         In order to recognize the ATmega328P-AU microprocessor as an Arduino chip, the proper bootloader must be burned onto the chip. To properly burn the bootloader, follow these steps:  
           
         1) Connect the MKII programmer to the AVR six pin header.  
         2) Open the Arduino IDE.  
         3) Under Tools, select Board: “Arduino Pro or Pro Mini”.  
         4) Under Tools, select Processor: “ATmega328P (3.3V, 8MHz)”.  
         5) Under Tools, select Burn Bootloader.
      2. **Program**  
         To properly program the ATmega328P-AU, follow these steps:  
           
         1) Connect the MKII programmer to the AVR six pin header.  
         2) Open the Arduino IDE.  
         3) Load the desired sketch.  
         4) Under Sketch, select Upload Using Programmer.
  1. **XBee S2C**

The XBee S2C radio module is a ZigBee based, serial communication module. The XBee S2C is rated for a maximum distance of 60’ indoors, can be set for bidirectional communication, and has built in collision avoidance. The XBee S2C can be configured by using a USB XBee adapter board and the Digi XCTU software.

* 1. **Wiring Harness**

The SmartLoo Wiring Harness is composed of a section of six conductor, 24 AWG wire spool. The six conductor wire section is soldered to the appropriate sensor on one end and a 2x3 position six pin male header on the other end.

* 1. **Enclosure**The current design of the SmartLoo Sensor Node utilizes commercially bought plastic enclosures. The enclosure is 3.8”x2.2”x0.8” in size and is milled out to allow for a 2x3 position six pin female header and a slide switch to be embedded into the walls of the enclosure. The 2x3 positon six pin male header soldered to the SmartLoo Wiring Harness can be inserted directly into the six pin female header.

**3.0 SmartLoo Bridge**

3.1 **ESP32 DevBoard v1.0**

3.1.1 **ESP32-WROOM-32 Module**  
The SmartLoo Bridge Node contains the ESP32-WROOM-32 chip, manufactured by expressif. This chip was a low-cost option for implementing the various secure connection types (WEP, WPA, WPA2-Personal, WP2-Enterprise) for maximum compatibility. It was configured to work with the BU (Guest) network. However, the device was tested at Columbia University and configured for WPA2-Enterprise, where it was successfully determined that enterprise support was possible.

This module can also be directly programmed with Arduino code, which means the board it exists on does not need to be connected to a board running a typical microprocessor like the ATmega328P-AU. It also supports necessary protocols like IPv4, IPv6, SSL, TCP, UDP, HTTP, FTP, and MQTT for maximum expandability.

3.1.2 **Power Supply**

This board supports 3.3V and 5V, which powers the on-board ESP32-WROOM-32 module and the attached XBee. The entire board is powered by a micro-USB cable with input of 5V and a maximum current of 1A.

3.1.2 **Programming**

This board supports a micro-USB for programming. This device was programmed using Arduino C using the Arduino IDE since the libraries for the microprocessor are updated frequently. However, this device could be programmed using C, but was scrapped as a result of lack of reference for this particular board. The following steps were taken:

1) Connect the device via USB.

2) Open the Arduino IDE.

3) Under Tools, make sure the following settings are chosen:

Board: “ESP32 Dev Module”

Flash Mode: “QIO”

Flash Frequency: “80 MHz”

Flash Size: “4 MB (32Mb)”

Upload Speed: “921600”

Core Debug Level: “None”

5) Click the circular arrow button (Upload) in the IDE.

3.2 **XBee S2C**

The XBee S2C radio module is a ZigBee based, serial communication module. The XBee S2C is rated for a maximum distance of 60’ indoors, can be set for bidirectional communication, and has built in collision avoidance. The XBee S2C can be configured by using a USB XBee adapter board and the Digi XCTU software. It is attached externally to the ESP32 using connectors to the 3.3V power, Ground, RX2 pin, and D4 pin (GPIO acting as TX2) on the ESP32 DevBoard.

3.3 **Enclosure**

The enclosure of the device is a square cube with dimensions 3.15in x 3.15in x 3.15in (8.0 cm x 8.0cm x 8.0cm) as it holds both the ESP32 DevBoard and the XBee. There exists a hole for the USB cable that connects to the device, and the other end is a USB-A male connector. The enclosure was chosen for its low-cost.