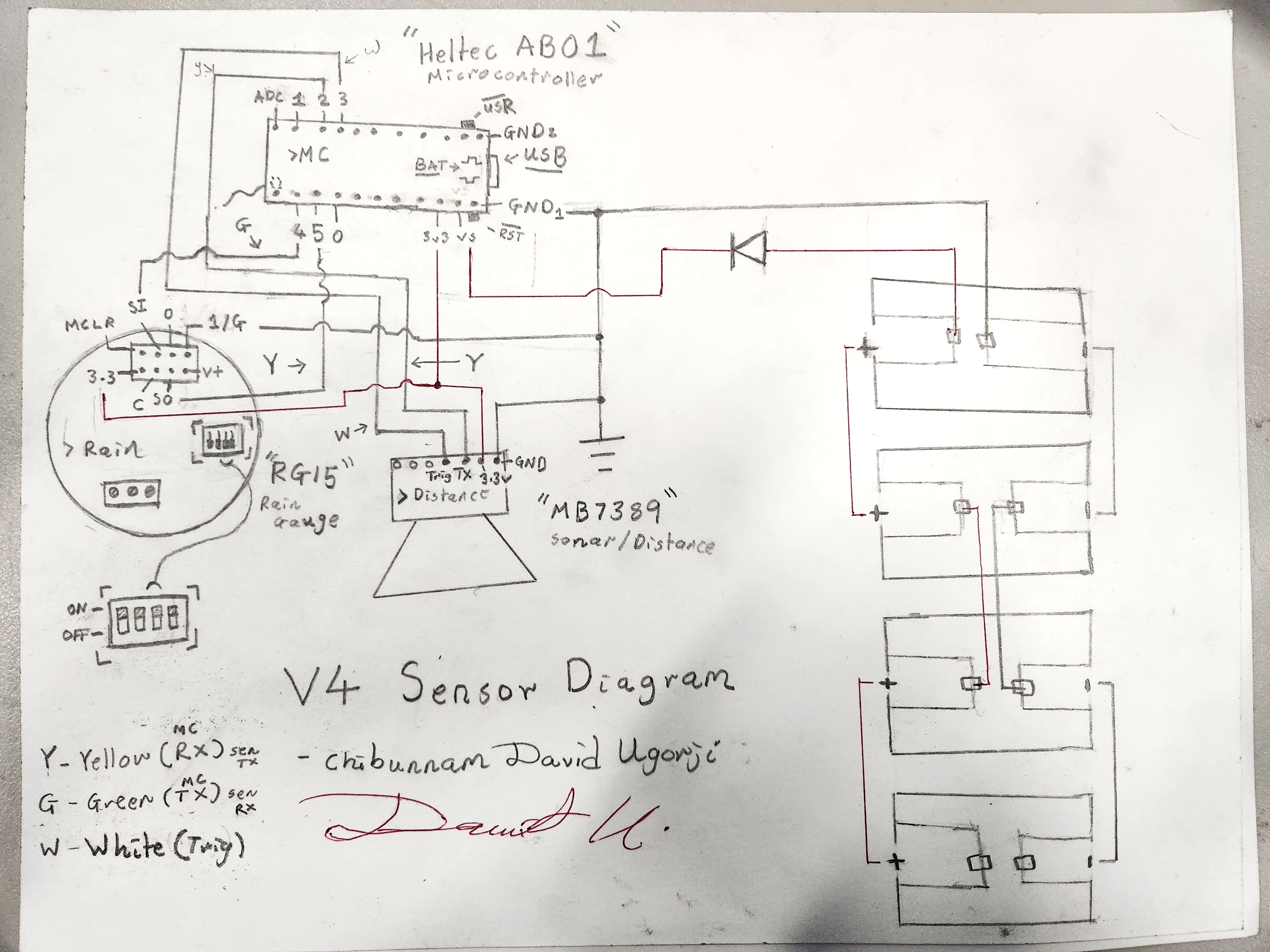
**Section 1 - Parts List**

| **Part** | **Description** |
| --- | --- |
| Heltec HTCC-AB01 | **Microcontroller.** The center of the V4 operation is the CubeCell AB01 development board developed by Heltec Automation. The board is fully Arduino compatible and runs on 3.3V Logic. Board includes the following relevant hardware:   * An antenna slot * 1 Serial USB C Port * 6 GPIO Pins (For Software Serial) * On Board Solar Charge Controller   On the software side the board includes:   * Software Serial * Built in LoraWan Libraries * On Board Watchdog   Datasheet, specs, and price can be found here:  <https://heltec.org/project/htcc-ab01-v2/> |
| Hydreon RG-15 | **Rain Gauge.** The Hydreon RG-15 collects rainfall data for the sensor unit. The sensor must be oriented upwards (glassy side up) to collect data. The RG-15 runs on 3.3V logic for this project but can be configured to run on 5V Logic. Lesser power to the sensor than configured results in incorrect readings, but greater power than configured will damage the sensor. The sensor can collect data in mm. or in in. and this can be configured by switches in the sensor (refer to Circuit Design). Sensor takes a read command and reports Accumulation, Event Accumulation, Total Accumulation, and IPH as a string.  Specs, and price can be found here:  <https://rainsensors.com/products/rg-15/>  Datasheet can be found here:  <https://rainsensors.com/wp-content/uploads/sites/3/2020/07/rg-15_instructions_sw_1.000.pdf> |
| MaxBotix MB7389 | **Ultrasonic Distance Sensor.** The MB7389 is an ultrasonic sensor that collects water level data for the sensor unit. The sensor collects data regardless of orientation, but is oriented with its bell towards the ground to collect water level. The MB7389 runs on 3.3V logic. Lesser power results in incorrect readings, but greater power will damage the sensor. The sensor automatically collects data when its trigger is pulled. Data is in mm. and is between 300mm to 5000mm.  Datasheet, specs, and price can be found here:  <https://maxbotix.com/products/mb7389> |
| Solar Panels (Unmarked) |  |

**Section 2 - Circuit Design**



***Figure: Hand Drawn Circuit***

The Circuit makes use of 4 GPIO pins to operate the sensors. The rain gauge’s “RS232 IN” (SI) is connected to GPIO 4 and its “RS232 OUT” (SO) is connected to GPIO 5. The distance sensor’s Trigger Pin (rightmost pin viewing from back) is connected to GPIO 3 and its RX Pin (second rightmost pin viewing from back) is connected to GPIO 2. The distance sensor’s Ground Pin (leftmost pin viewing from back) and the rain gauge’s Ground Pin (GND) are connected to a common ground. The distance sensor’s 3.3V Pin (second leftmost pin viewing from back) and the rain gauge’s 3.3V Pin (GND) are connected to the 3v3 pin on the dev-board.

Solar panels are connected in parallel with the ground wire being connected to the common ground, and the hot wire connected with a diode to the VS pin on the dev-board.

**Section 3 - Challenges**

New Micro-Controller

Previous sensor unit models (V1, V2, and V3) used an Adafruit Feather M0 SAMD board which had multiple Serial UARTS and a different implementation of Long Range RF Communications (Lora). Because the V4 uses the AB01, the design for the code and circuit had to be entirely different. One of the biggest challenges to development was the sparse documentation for the AB01 dev-board. Some code was constructed using examples provided by the CubeCell arduino examples library. However, most insights regarding faced errors were gleaned with painstaking effort from community boards and example code for unrelated projects.

Connectivity Issues

One of the major challenges to the project was its connectivity to the LoraWan network as well as its effect range to the in-house network gateway. The setup for the previous versions was a connection to The Things Network (TTN) through the in-house gateway. However, when testing transmission at range, both the V3 and V4 did not achieve more than 0.3 miles transmit range. The primary assumption was that the sensors were having connectivity problems. After some research and more testing, the discovery of low area coverage by TTN proved to be the problem. The sensors were physically distant from the gateway’s effective range and had little of a mesh network in the testing area to fall back onto. The project was henceforth moved onto the Helium Console network which had greater coverage.

Power Consumption & Operation Interrupts

Power consumption was a major challenge. Previous versions (V1 and V2) did not include the rain gauge, but only the distance sensor which was less complex in connection and power management. The addition of the rain gauge in the V3 design added more power drain to the unit. Although the V3 used 5V logic on the rain gauge, the drain of running both sensors was still a significant issue in the V4 design. Compounded on this was the problem of frequent program hangs in which the board would freeze and neither read nor send any data. Both problems were rectified with the implementation of low power mode and the internal watchDog library. Low power mode would reduce board power while the sensor was “idling” and the watchDog would wake up the board in the event of a program hang.

**Section 4 - Code and Configuration**

Setup

The majority of the AB01’s functionality comes from hardware libraries integrated into Arduino’s base functionality at installation. Therefore setup for the board includes installation of the CubeCell hardware library as well as download of the CP210x USB to UART drivers. The drivers help recognize the board when connected to a computer. The board must be connected with a USB C to USB A data transfer cable.

Code Sectors

The code for the sensors is detailed in several cpp and h files that each handle different aspects of sensor function. The code is split into 4 “sections” namely:

* A main body (V4\_lora.ino, V4\_lora.h) that details board setup and looping behavior as well as implements low power and watchDog.
* A section for lora (lora.cpp, lora.h, secrets.h) that details connection setup, and handles uplinks and downlinks.
* A section for sensor behavior (mcu\_functions.cpp, mcu\_functions.h) that details functionality such as individual sensor reads as well as data parsing.
* A section for utilities (utils.cpp, utils.h) that provides lower-level functions such as the instantiation of softwareSerial ports for sensors or payload encoding for lora transmission.

Further descriptions can be found in the code itself.