HabPi: An Open Source Extensible Framework for High Altitude Balloon Data Collection

Robert Lowe*
Maryville College

One of the obstacles facing beginning high altitude balloonists is the cost of the myriad sensors required to acquire usable data. In addition to this problem, aligning sensor data after a flight can prove tricky. Finally, there is a harsh reality we all face. Trees and other tall structures pose a real risk to the recovery of any HAB payload.

HabPi is an open source project which attempts to address these problems. This is a sensor array which is based on a Raspberry Pi, a Sense HAT, and several one-wire sensors. The sensors are all synchronized, so data registration is not a problem. HabPi also acts as a wifi access point which makes data retrieval possible even when the stack is trapped in a tree (or is otherwise inaccessible). HabPi is a simple extensible framework which allows users to add additional experiments beyond the default setup. The default setup provides temperature, pressure, magnetometer, gyroscope, accelerometer, and video imaging all for less than $200.00.

I. Introduction

The HabPi project is both an open source software package and a recommended hardware configuration. The aim of this project is to provide a simple and extensible high altitude sensing system which is both economical and robust. The default configuration of the HabPi system results in a platform capable of capturing the following data:

- Temperature
- Humidity
- Barometric Pressure
- Magnetometer Data
- 3-Axis Accelerometer Data
- 3-Axis Gyroscope Data
- Still Photos / Video

In addition to these basic data points, the system provides many opportunities for expansion. For example, there have been flights carrying cosmic ray detection experiments using a similar hardware setup.

In addition to its low cost, this system provides a central point of contact for all sensor data. This means synchronization of recorded data is trivial as all data points are recorded with a common clock. In fact, data recording procedures can be crafted which concurrently capture data points the user is interested in comparing, thus eliminating the challenges of synchronizing data across disparate sensors.

A final advantage is flexible data recovery. If the payload can be recovered, then the on board computer is sufficient to process the information obtained during the flight. If the payload is not recoverable, the HabPi system provides a mechanism for wireless data recovery. As long as a user can get to within a few hundred feet of the payload, the data will always be recoverable and usable. (Provided, of course, the payload lands intact.)

A. Motivation

The primary motivation for the creation of the HabPi system was cost. When presented with the prospect of starting a ballooning program, it is easy to become overwhelmed with the choices of available sensors and the costs of their given platforms. Also, many of these platforms necessitate the use of proprietary software which can create a situation where experimental efforts are stymied by lack of control.

Another motivation for the creation of HabPi in its present form was the loss of a payload in November of 2016. The first HabPi payload launched by a joint effort between Maryville College and Pellissippi State Community College came to rest at the top of a very large tree in Pisgah National Forest. While the payload was sighted, and indeed visited on five

*Assistant Professor, Division of Mathematics and Computer Science, Maryville College, 502 East Lamar Alexander Parkway, Maryville TN 37804

Item	**Approximate Cost**
Raspberry PI 3 Model B.	\$35.00
Sense HAT	\$30.00
Raspberry Pi Camera V2	\$30.00
32GB Micro SD Card	\$15.00
3 × DS18B20 Digital Thermometers	\$9.00
4.7kΩ 1/4 W Resistor	\$0.10
30 Row self-adhesive breadboard	\$5.00
4400 mAh USB Battery (Cell Phone Charger)	\$10.00
20 pack of Male/Female Jumper Wires	\$3.00
300mm Ribbon Cable for Pi Camera	\$2.00
Total	**\$139.10**

Table 1 Parts List

occasions, all of the precious data from the flight was trapped some 65 feet above the ground. The addition of wireless download capabilities was a natural result of the frustration from this all too common event.

B. Educational Use

In addition to its use in normal stratospheric flight experiments, HabPi has also proved to be very useful in education. It was developed and refined using input from students at Maryville College and Pellissippi State Community College. In addition to college students, HabPi payloads have also been constructed by elementary, middle, and high school students from Concord Christian School. Boy Scout Troop 255 also built and flew a HabPi payload. Students from each age group were very enthusiastic to construct the payload, and all were able to successfully construct the HabPi payload.

In addition to collecting stratospheric data, the default setup of a HabPi system works well as a ground based weather station. Curricula including weather experiments culminating in a stratospheric flight are currently under development, and will hopefully be the topic of a future paper.

C. Organization of this Paper

The remainder of this paper is organized as follows. First, the construction of the HabPi payload is outlined. This includes both the sensor hardware and the recommended box dimensions. Following the construction section is a detailed description of the HabPi software, including both setup and operation. The final two sections of the paper present flight data and some concluding remarks about the performance of the HabPi system.

II. Construction

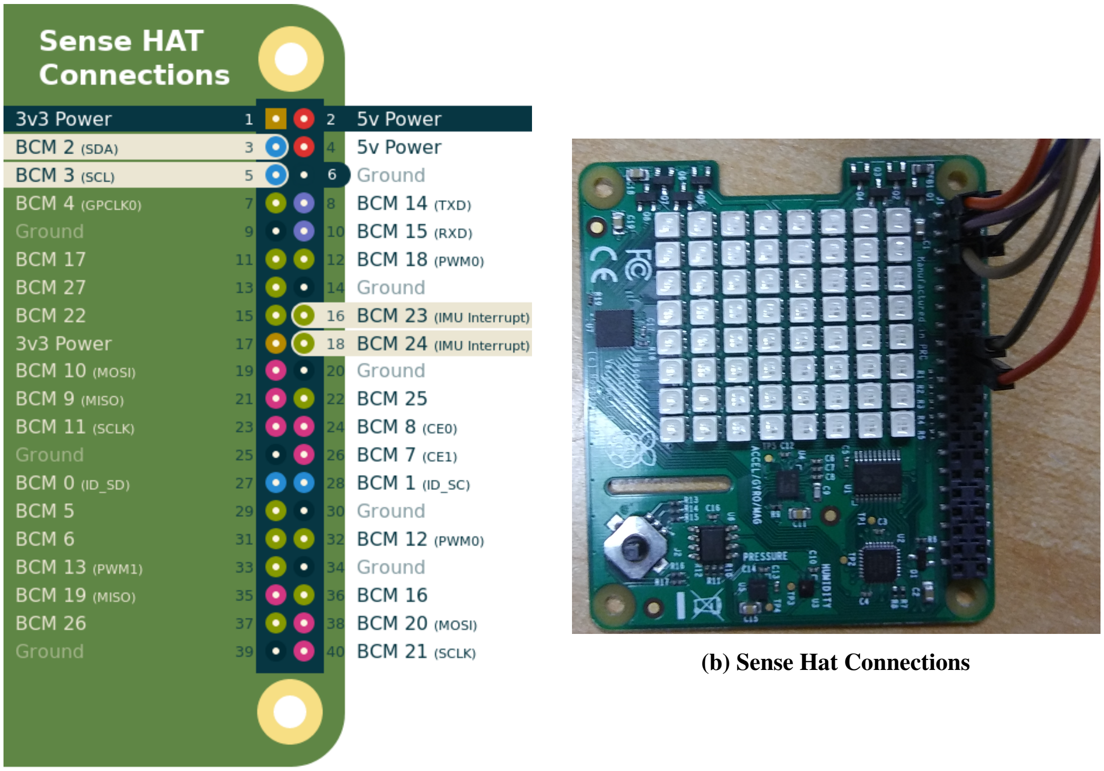
The HabPi setup consists of four principal components:

1) 1 Raspberry Pi version 3 Computer [1]
2) 1 Sense HAT board [2]
3) 1 Raspberry Pi Camera V2 [1]
4) 3 DS18B20 1-Wire Temperature Sensors [3]

A complete parts list for the electronics, along with approximate cost, can be found in Table 1.

The Raspberry Pi computer is designed to be a friendly Linux based system which is easily programmable. Of particular interest is the presence of both USB ports and an array of programmable GPIO (General Purpose IO) pins. This allows the Raspberry Pi to serve as both a driver and processor for sensors using a variety of communication technologies. In addition to its wired connections, the Raspberry Pi version 3 also provides wireless connectivity in the form of bluetooth and wifi. The wifi adapter can be configured to act as an access point, allowing the Raspberry Pi to function as the hub of a wireless network.

The Sense HAT board is an array of sensors which was designed for the Astro Pi mission [4]. A device containing


Sense HAT Connections
3v3 Power 1 2 5v Power
BCM 2 (SDA) 3 4 5v Power
BCM 3 (SCL) 5 6 Ground
BCM 4 (GPCLK0) 7 8 BCM 14 (TXD)
Ground 9 10 BCM 15 (RXD)
BCM 17 11 12 BCM 18 (PWM0)
BCM 27 13 14 Ground
BCM 22 15 16 BCM 23 (IMU Interrupt)
3v3 Power 17 18 BCM 24 (IMU Interrupt)
BCM 10 (MOSI) 19 20 Ground
BCM 9 (MISO) 21 22 BCM 25
BCM 11 (SCLK) 23 24 BCM 8 (CE0)
Ground 25 26 BCM 7 (CE1)
BCM 0 (ID_SD) 27 28 BCM 1 (ID_SC)
BCM 5 29 30 Ground
BCM 6 31 32 BCM 12 (PWM0)
BCM 13 (PWM1) 33 34 Ground
BCM 19 (MISO) 35 36 BCM 16
BCM 26 37 38 BCM 20 (MOSI)
Ground 39 40 BCM 21 (SCLK)
PRESSURE
HUMIDITY
ACCEL/GYRO/MAG

(b) Sense Hat Connections

(a) Minimal Raspberry Pi Sense HAT Connections [9]

Fig. 1 Wiring the Sense Hat

both a Raspberry Pi and a Sense HAT was sent to the international space station, and children from around the world contributed experiments as part of a contest where the winners had their experiments carried out aboard the ISS. The Sense HAT provides the following features:

- Two Digital Thermometers (−40°C to +120°C) [5] and (−30°C to +105°C)[6]
- Barometric Pressure Sensor (260 to 1260 hPa) [6]
- Relative Humidity Sensor (0% to 100%) [5]
- 3-Axis Magnetometer (±16 gauss) [7]
- 3-Axis Gyroscope (±2000 dps) [7]
- 3-Axis Accelerometer (±16g) [7]
- Joystick for Input
- RGB LED Array for Output

As can be seen in the above list, the temperature ranges provided by the Sense HAT are not sufficient for recording stratospheric temperatures. Also, because the Sense HAT electronics generate heat during normal operations, the temperatures read by the Sense HAT will tend to be higher than the ambient temperature [8]. To combat this, the standard HabPI setup uses 3 DS18B20 1-wire thermometers which have an advertised temperature range of −55°C to +125°C [3].

A. Wiring the Sense HAT

In its standard configuration, the Sense HAT has a connector which fits over all of the Raspberry Pi's GPIO pins. This is undesirable for two reasons. First, this means no further GPIO hardware can be added. Second, and most importantly, the heat from the Raspberry Pi will dramatically increase the temperature value read by the Sense HAT. Fortunately, this connector can be removed by simply pulling it out of the Sense HAT. To wire the Sense HAT, only seven connections are actually necessary. These connections are shown in Figure 1a, with Figure 1b showing the corresponding orientation of the Sense HAT. These connections are made using Male/Female jumper wires.

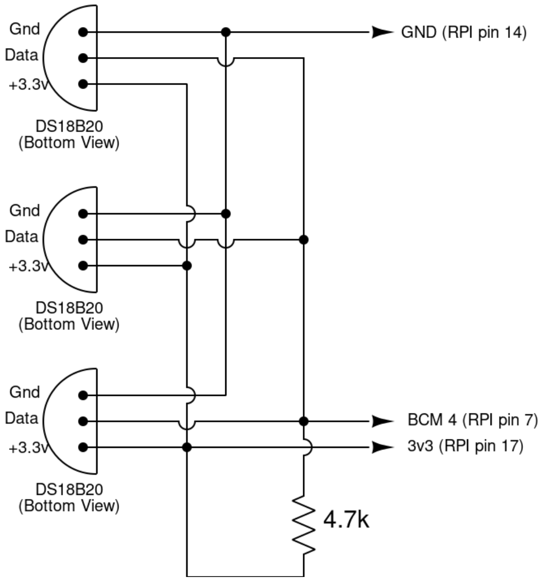
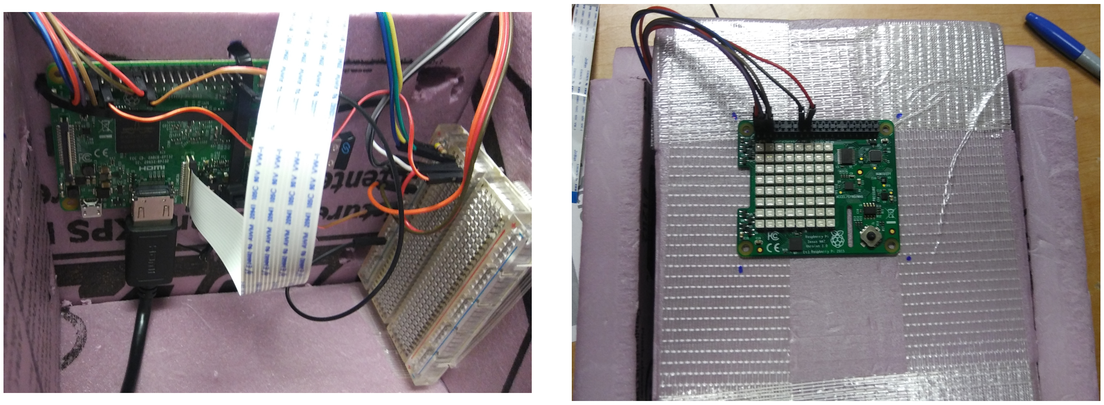


Fig. 2 DS18B20 1-Wire Circuit



(a) Interior with Mounted Computer and Breadboard

(b) Top Showing Mount Points for the Sense HAT

Fig. 3 Mounting the Electronics

B. Wiring the 1-Wire Sensor Array

The DS18B20 1-wire sensors share a common data line which is connected to a pull-up resistor. Any number of 1-wire sensors can share this single pull-up resistor. The power and ground of each of the sensors is connected to the 3v3 and GND line on the Raspberry Pi respectively. This circuit is built on a solderless breadboard according to the schematic shown in Figure 2. Jumper wires are used to make all of the connections to the raspberry pi.

C. Box Dimensions and Mounting

The recommended payload enclosure has exterior dimensions of 8” × 8” × 8” and interior dimensions of 6” × 6” × 6”. This provides ample room to mount the base setup and to allow access to the display and power ports of the Raspberry Pi. This also provides enough space to add additional experiments. Of course, the enclosure can be any dimensions the user wishes, so long as the wires can be made to reach. The self adhesive breadboard is affixed to the inside of the box, and the battery is mounted to the bottom. The suggested layout of the inside of the box is shown in Figure 3a.

In order to isolate the Sense HAT from the heat generated by the other components, it is mounted to the lid of the box on the outside. The easiest way to mount the Sense HAT is by passing wire through the foam at the corners and twisting it on the other side of the lid. Figure 3b shows the positioning of the Sense HAT along with markings for the wire mount holes.

The camera is mounted in the same fashion on the outside of the box, and can be positioned wherever the user desires. Also, tape is applied to the ribbon cable to secure it to the outside of the box. One advantage of using the Raspberry Pi camera in this fashion is that it does not require large holes to be cut into the insulation of the payload box.

The temperature sensors are all connected to jumper wires and left to dangle freely. The suggested way to fly these sensors is with two on the outside of the box and one on the inside of the box in order to measure the efficacy of the insulation*.

III. HabPi Software

The HabPi software is a series of python scripts which form a framework for collecting data from the sensors. The software is released under the GPL3 license, and is available at https://github.com/pngwen/habpi [10].

HabPi is designed to run on top of Raspbian Stretch, though it can be used with other distributions on the Raspberry Pi. HabPi creates a hierarchy of folders in the home directory of the pi user. These are as follows:

- **habpi/** The HabPi distribution itself.
  - **HabPi/** The HabPi python module.
  - **data/** The collected data from HabPi.
  - **scripts/** Utility scripts for the HabPi system.
  - **experiments/** The experiment scripts to be executed when HabPi collects data.
  - **examples/** Example Experiment Scripts
    - **flash.py** A script which causes the LEDs on the Sense HAT 3 hours after activation. (Useful for Recovery)
    - **senseRec.py** A small script to capture all of the data from the Sense HAT in a format that can be played back using the Sense HAT emulator.
    - **timeLapse.py** A time lapse photography script. This captures one image per second. The delay can be adjusted by changing the code.
    - **video.py** A video recording script. It captures 10 minutes of video per file and then rotates to another file. This is to decrease the chance of losing video data during power failure.
    - **temp.py** A script which logs temperatures from all five thermometers into a CSV file. It produces one reading per second.

Whenever HabPi records data, it creates a new directory under the data folder named <*date*>-<*time stamp*>. Thus any number of recordings can be stored, space allowing.

A. Raspberry Pi Setup

The first step in setting up HabPi is to prepare the Raspberry Pi. To do this, install Raspbian and then set up the localization options. Change the default password, and then activate both the camera and ssh interfaces. Finally, install *hostapd* and *dnsmasq*. These last two are needed to allow the Raspberry Pi to act as a wifi access point.

To complete the setup, we add the following line to the crontab of the raspberry pi user:

```
@reboot /home/pi/habpi/scripts/start
```

B. Setting up HabPi

The primary configuration for HabPi is located in the file `HabPi/config.py`. The default configuration is to have HabPi act as an access point with the SSID of "SkyNet" and the password of "Terminat0r". This can be changed, however. When running as an access point, no wifi internet connection is possible, but the Raspberry Pi can still be connected to the internet via wired ethernet. Setting the `accessPoint` variable to `False` will cause the system to use client mode wifi instead.

C. Creating Experiments

The final step in preparing to fly is to create the experiments. Any file ending in `.py` contained in the experiments folder will be executed in its own thread when data recording begins. What these experiments do is completely free-form, but a good first step would be to copy examples into the experiments folder. It should be noted that the video and time

*This can be a source of friendly competition when working with groups of grade-school students! The winner is the group who's box stays warmer.

lapse examples are mutually exclusive as the system cannot do both at once due to the limitations of the Raspberry Pi camera.

Experiments should import the "HabPi" module. This module provides the following information:

HabPi.sense The Sense HAT object. This should be used instead of a manual connection so that the experiment can use the emulated Sense HAT if it is configured.

HabPi.dir The full path to the data directory. The experiment should record data here.

HabPi.sensors Sensor states. This is a dictionary to which each experiment can publish information and allows for inter-script communication.

D. HabPi User Interface

When the system starts, HabPi will start automatically. The LEDs will begin flashing a question mark "?", as a prompt. There are then two ways to interact with HabPi.

1. Sense HAT Joystick

Pressing up on the joystick will cause HabPi to enter "Date Set" mode. This will allow the date to be set using the joystick one digit at a time. Pressing right on the joystick allows the user to enter the current time. Pressing left on the joystick will display the current date and time. These are necessary because the Raspberry Pi lacks a real time clock and needs to have its system time reset every time it starts. (Adding an RTC is an interesting and simple project, but by default there is not one present.) Pressing down on the joystick will cause HabPi to count down. It will display the sequence 3, 2, 1. Pressing the joystick during the count down will cancel the recording and return to the prompt. Otherwise HabPi will flash an "R" to indicate that it has started recording data.

2. Wireless Access Point

The HabPI wireless access point provides another way to interact with the device. A computer can connect to this access point, and then use either ssh or sftp to contact the server at address 192.168.1.1. This allows for scripts to be started, and for data to be downloaded remotely.

IV. Performance and Flight Data

The data in this section are from a stratospheric flight which took place on October 7, 2017. The balloon was released from Athens, TN at approximately 14:00 UTC and the payload was recovered in Knoxville, TN four hours later. The balloon reached a peak altitude of 82,789 feet according to the Iridium satellite tracker which was part of the payload stack.

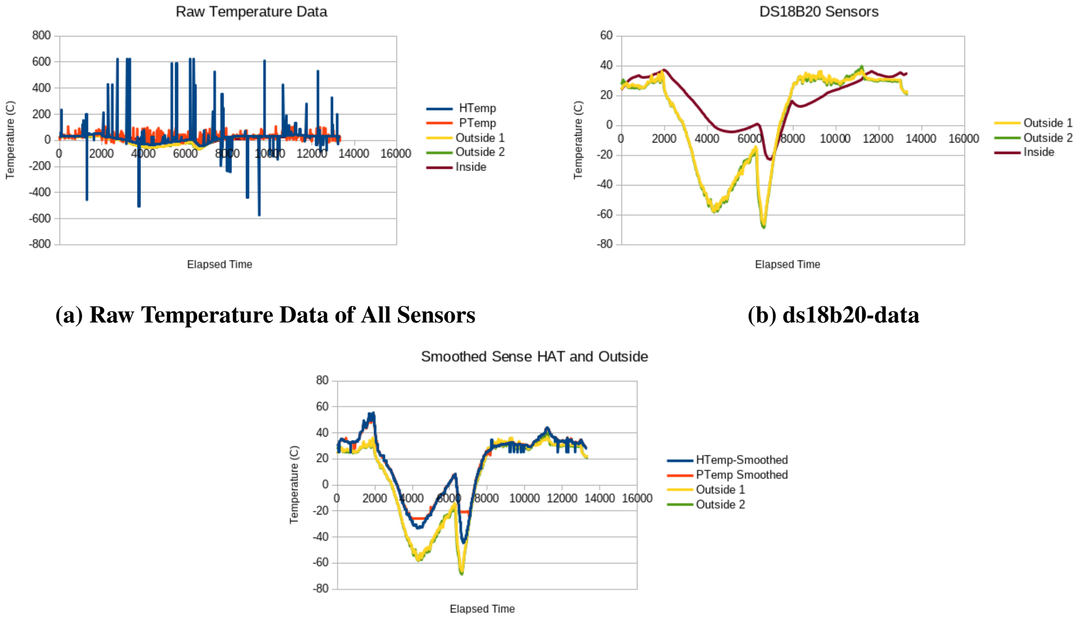
A. Temperature Data

The temperature data from this flight shows one of the key challenges to using Sense HAT data. The Sense HAT readings tend to be very noisy, as can be seen in Figure 4a. The DS18B20 sensors, on the other hand, produce very clean information, as can be seen in Figure 4b. Fortunately, smoothing the Sense HAT data is fairly simple. The filter applied in Figure 4c is accomplished by simply excluding any data point which differs by more that 5°C from the previous data point.

As is predicted by the documentation of the Sense HAT, these temperatures are warmer than the temperatures captured by the DS18B20 sensors. Also, note that the DS18B20 sensors were painted white to reduce solar heating, but the Sense HAT sensors cannot be painted as this will interfere with their pressure and humidity readings.

B. Other Sensors

The other sensors in the Sense Hat are not quite as noisy as the temperature sensors. The pressure sensor has occasional spikes, but in general it tracks well for the duration of the flight. In this flight, the pressure altitude reached by the balloon at its peak was 85,388 feet.



(a) Raw Temperature Data of All Sensors

(b) ds18b20-data

(c) External Temperature data with Smoothed Sense HAT data

Fig. 4 Temperature Data

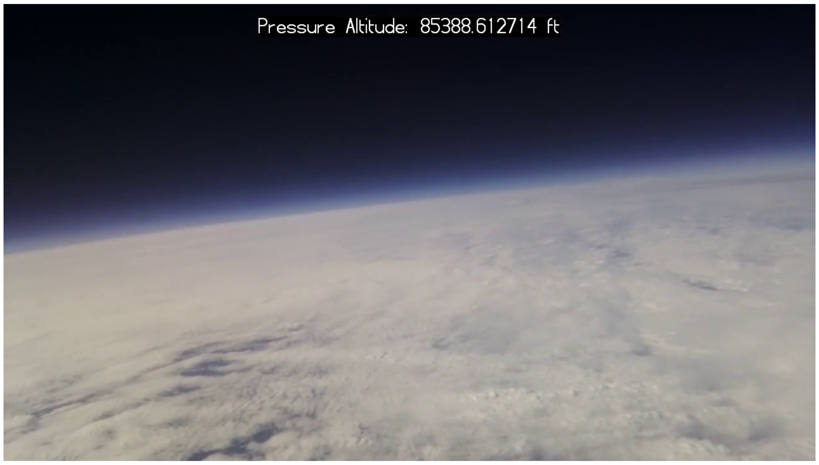


Fig. 5 Peak Altitude with Overlay

V. Conclusion

The HabPi system provides an open framework for high altitude sensing based on readily available components. It is simple to build, so much so that it can be assembled by children, but it is also robust enough to produce usable data. Because the software is completely open source, the user can alter all of the data collection routines to suit their needs. For example, one such extension is to overlay the current pressure altitude on the captured video files, as seen in Figure 5. The Raspberry Pi itself offers additional 1-wire capabilities as well as many available GPIO pins and USB ports.

Future work will include the development of curricula for using HabPi in classrooms, as well as more experiments driven by this framework.

VI. Acknowledgments

The author would like to acknowledge Sarah Graham and the students of Pellissippi State Community College for providing multiple opportunities to build and fly HabPi payloads. Special thanks to the children of Concord Christian School and Troop 255 for building early versions of HabPi payloads are also in order.

References

[1] `https://www.raspberrypi.org`, 2017. Accessed: 2017-10-09.

[2] `https://www.raspberrypi.org/documentation/hardware/sense-hat/README.md`, 2017. Accessed: 2017-10-09.

[3] *DS18B20 Programmable Resolution 1-Wire Digital Thermometer*, Maxim Integrated Products Inc., 2013. Revision 042208.

[4] Editors, M., *Experiment with the sense hat sense the real world with your Raspberry Pi*, Raspberry Pi, 2016.

[5] *HTS221 Capacitive digital sensor for relative humidity and temperature*, ST Microelectronics, 2016. Revision 4.

[6] *LPS25H MEMS Pressure sensor*, ST Microelectronics, 2016. Revision 5.

[7] *LSM9DS1 iNEMO inertial module: 3D accelerometer, 3D gyroscope, 3D magnetometer*, ST Microelectronics, 2015. Revision 3.

[8] `https://pythonhosted.org/sense-hat/`, 2017. Accessed: 2017-10-09.

[9] `https://www.pinout.xyz`, 2017. Accessed: 2017-10-09.

[10] "HabPI Code Repository," `https://github.com/pngwen/habpi`, 2017.