Window Sensors to Advance Bird-Friendly and Energy Saving Building Design Strategies on UBC Vancouver Campus

PL-68

BENJAMIN POWELL
CRYSTAL LI
HUAWEN LI
MOHAMED SALAH
RYOTARO HOKAO

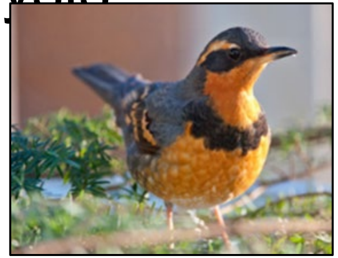
Background Problem

- Bird collisions with windows are one of the top sources of bird mortality
  - 10,000 bird collisions occur on UBC per year
  - Birds cannot see the transparent windows
- UBC's Green Building Action Plan set a target for 2035: all new and rebuilt buildings must achieve low energy consumption and emissions

Project Goals

- Build upon bird collision detector designed by past Capstone
- Detect bird-window collisions
- Record information related to bird strike (date, time, location, temperature) *New*
- Calculate heat flow rate of windows *New*
- Automatically upload data to database and dashboard that can be exported for further analysis






Target User

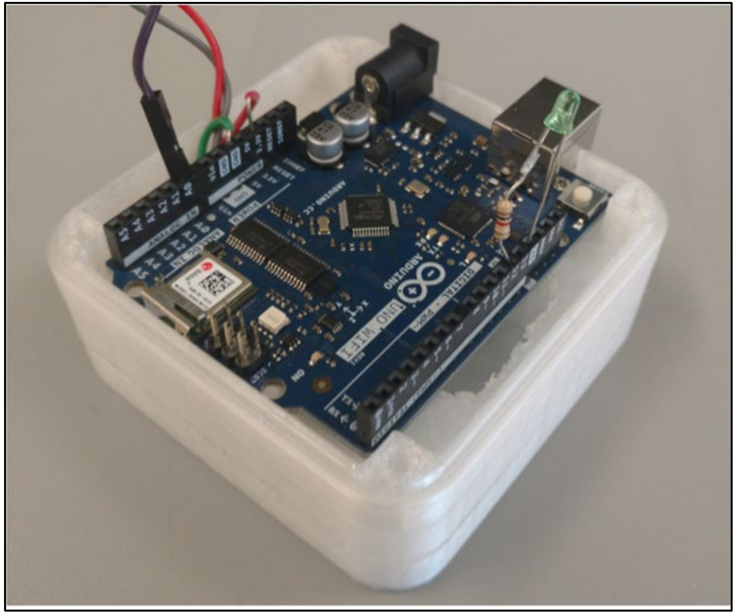
Researchers

- Help determine the possible factors behind bird collisions (eg. time, temp)
- Examine the thermal insulation of windows to monitor energy efficiency

Developers

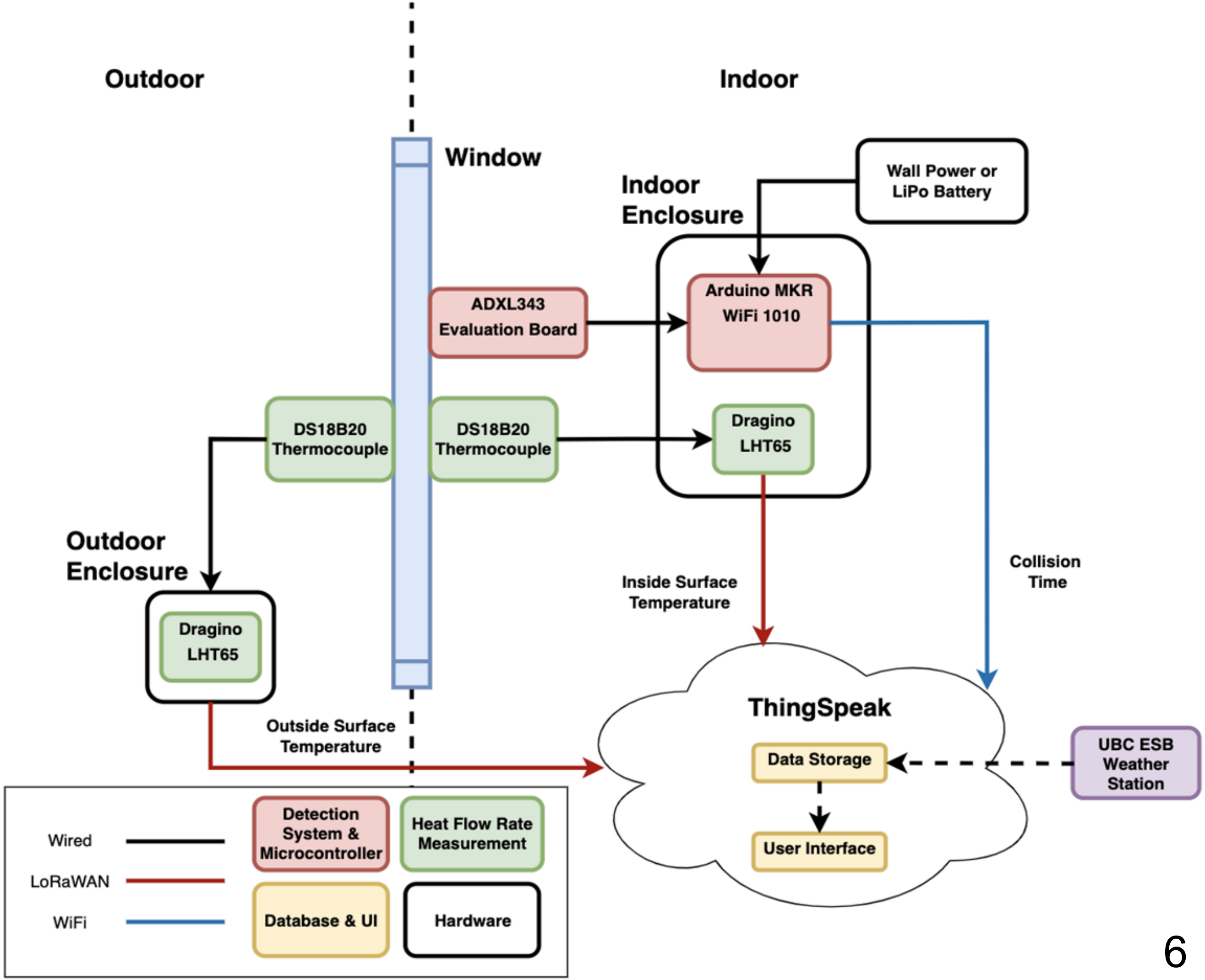
- Promote building compliance with the GBAP, the UBC Climate Action Plan 2030 (CAP 2030), and Bird Friendly Building Design Guideline

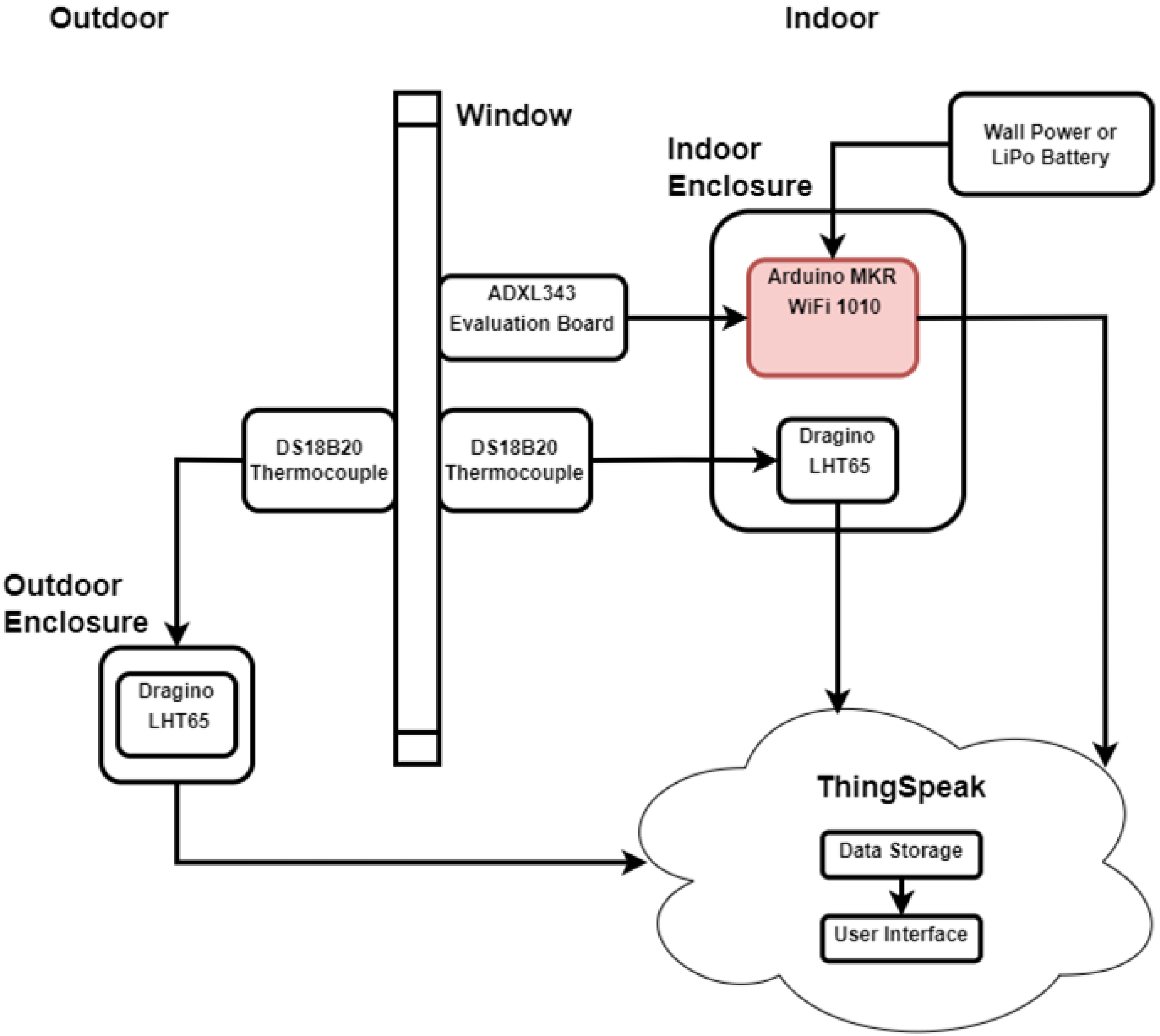
Previous Capstone Design



- $80 per window
- Finite State Machine for task scheduling
- Only monitor bird-window collisions
- Cayenne as data storage and UI

System Architecture





Microcontroller system

Microcontroller System

MKR Wi-Fi 1010

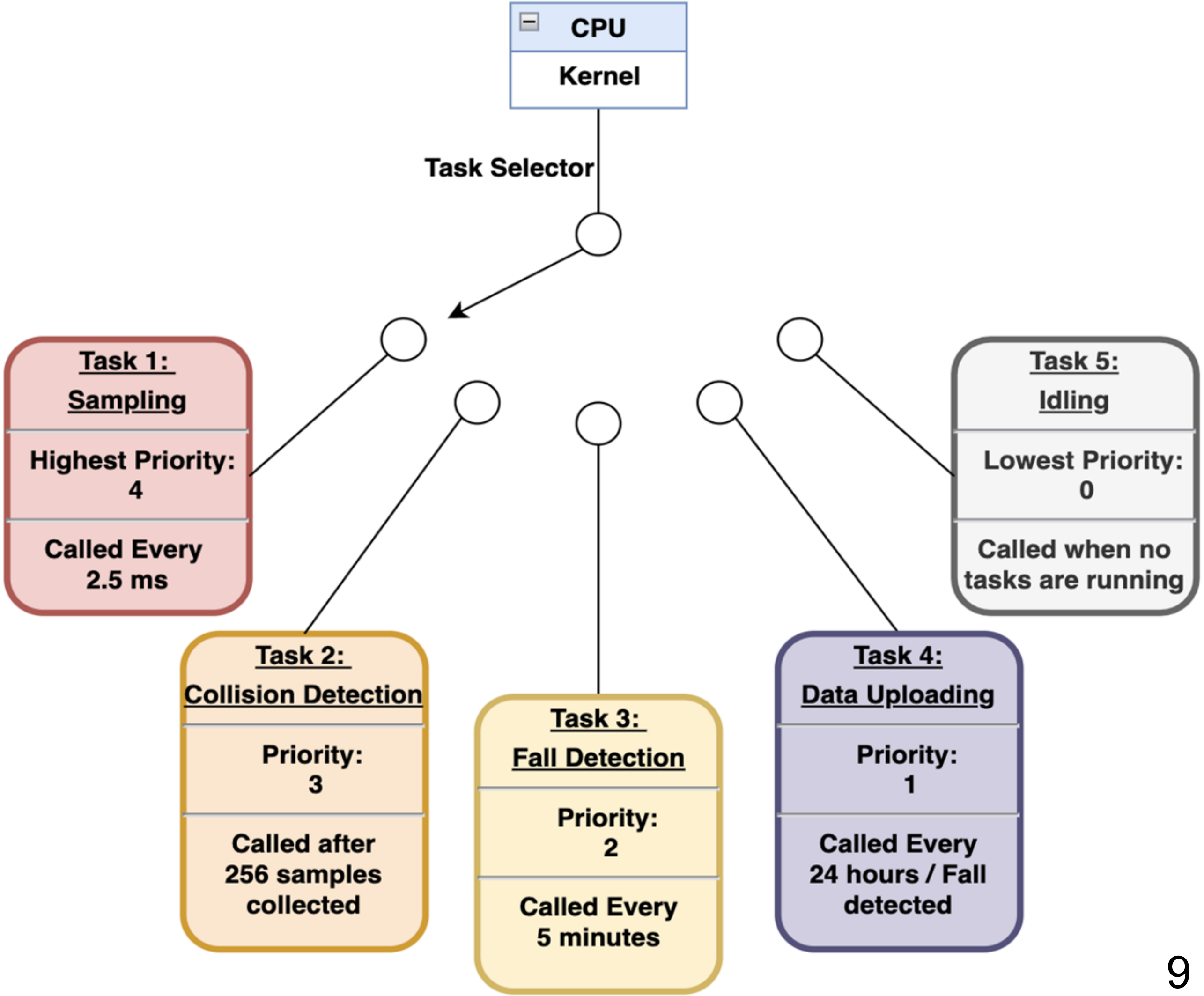
- WiFi Connectivity
- Enough digital pins
- Power socket module
- Ample flash memory capacity (256KB)
- Cost-effective (CA$52.12)
- Compact size and weight (32g)(25mm x 61.5mm)

FreeRTOS

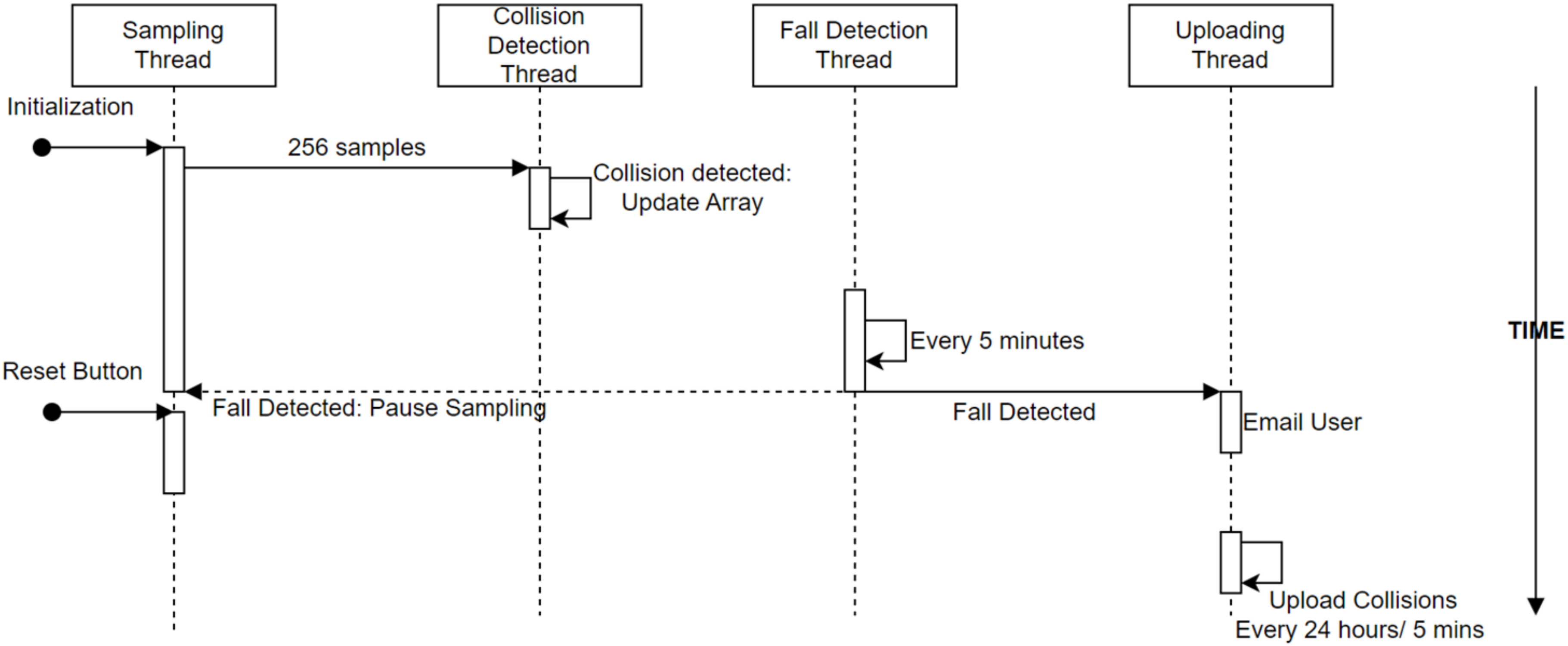
(New Feature)

- Open-source, real-time operating system
- Offers a simple, efficient, and lightweight solution for managing tasks, time, and resources

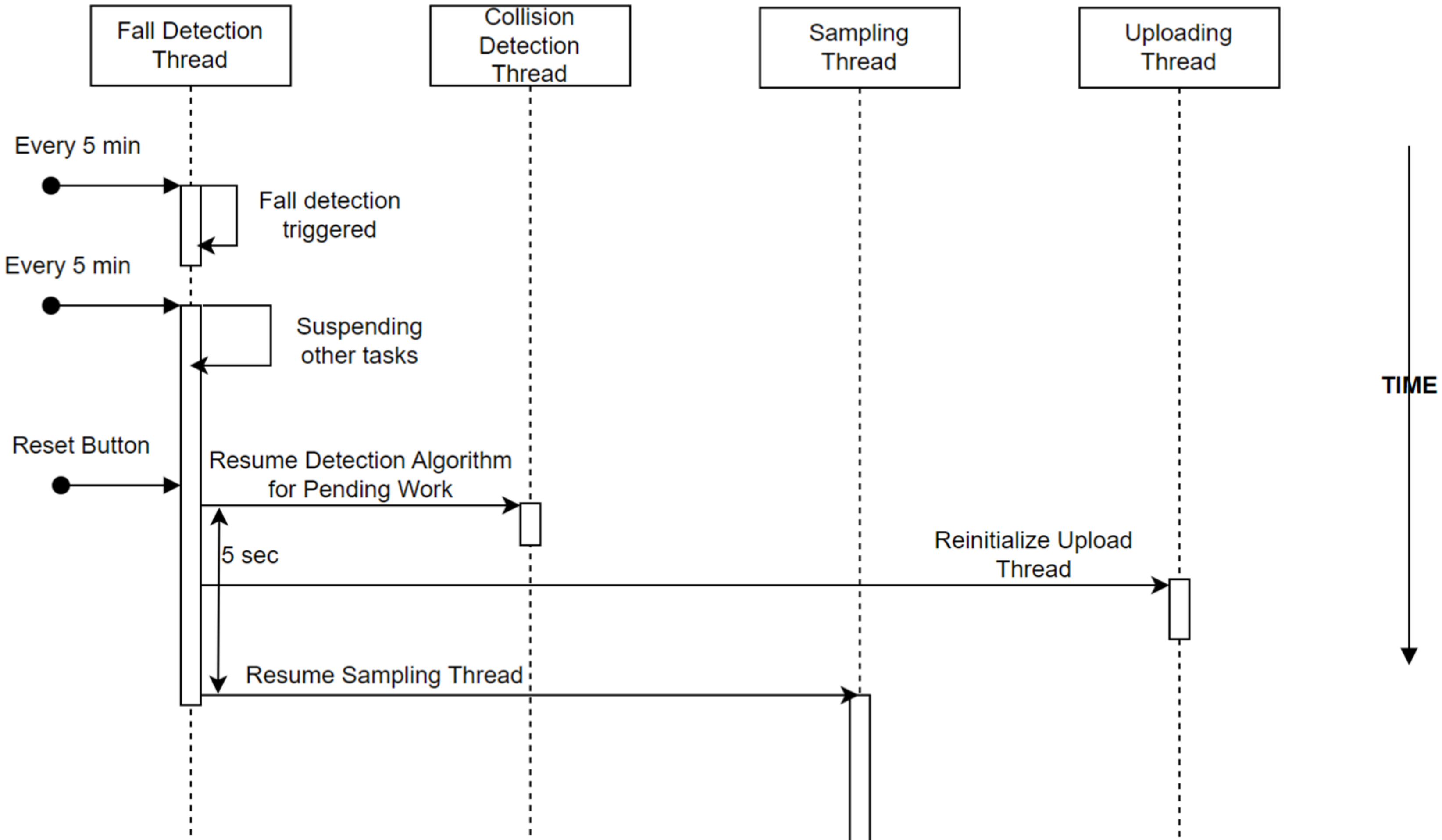
Microcontroller System – Threading



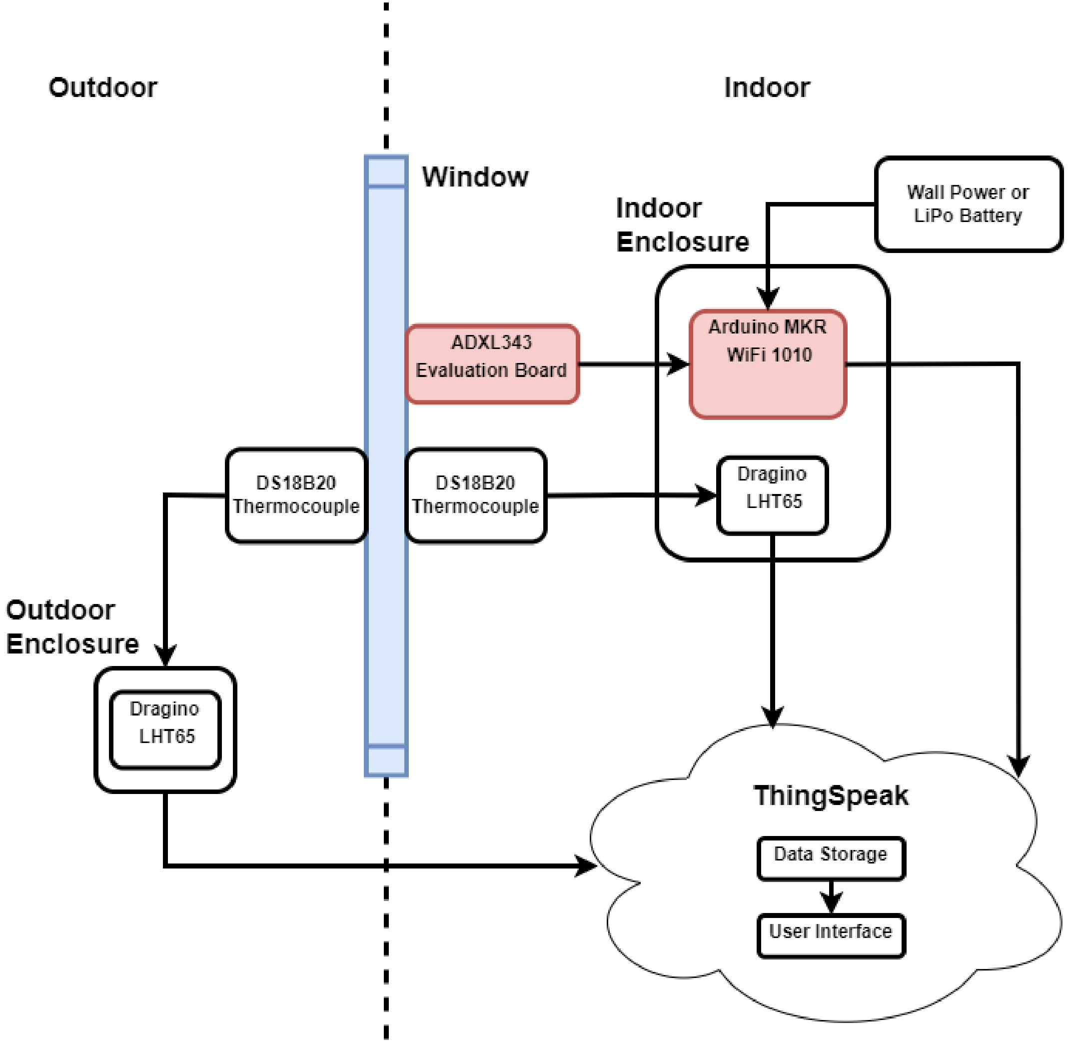
Thread Timing Diagram



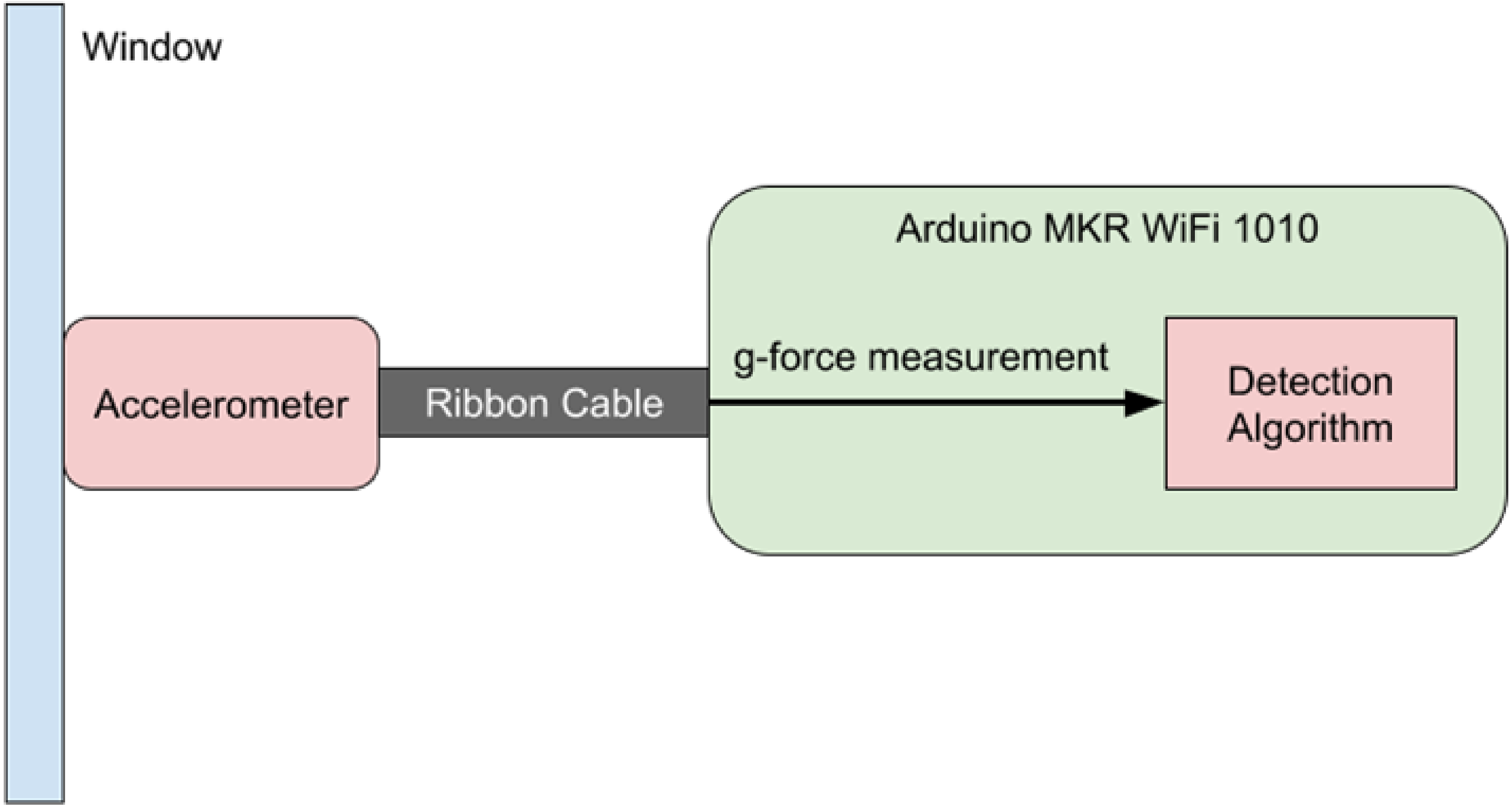
Fall Detection Timing Diagram



Detection System



Detection System



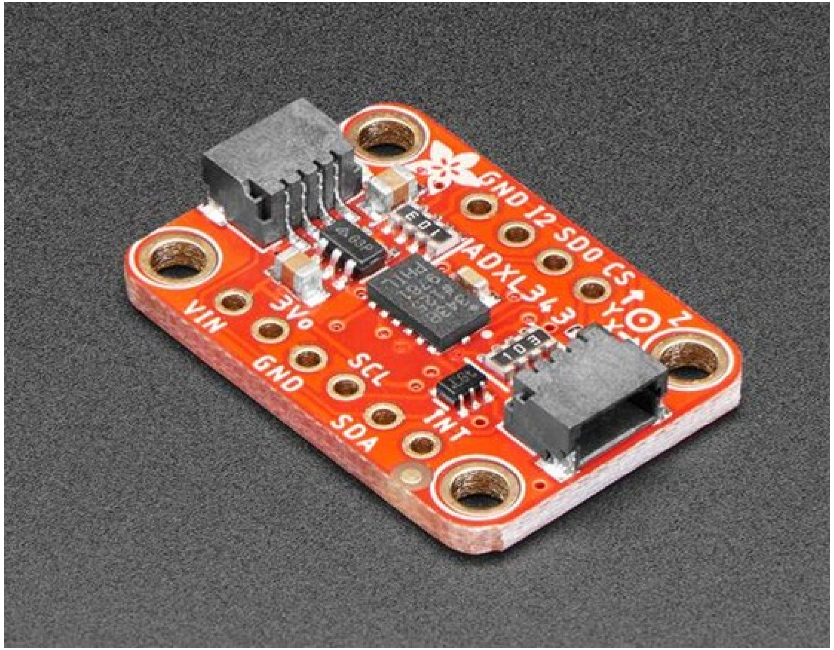
Detection System – Accelerometer

ADXL343 Evaluation Board:

- Digital 3-axis MEMS accelerometer
- Low cost (CA$8.72)

- +/- 16g measurement range
- 3.9 mg sensitivity
- Communicate via I2C

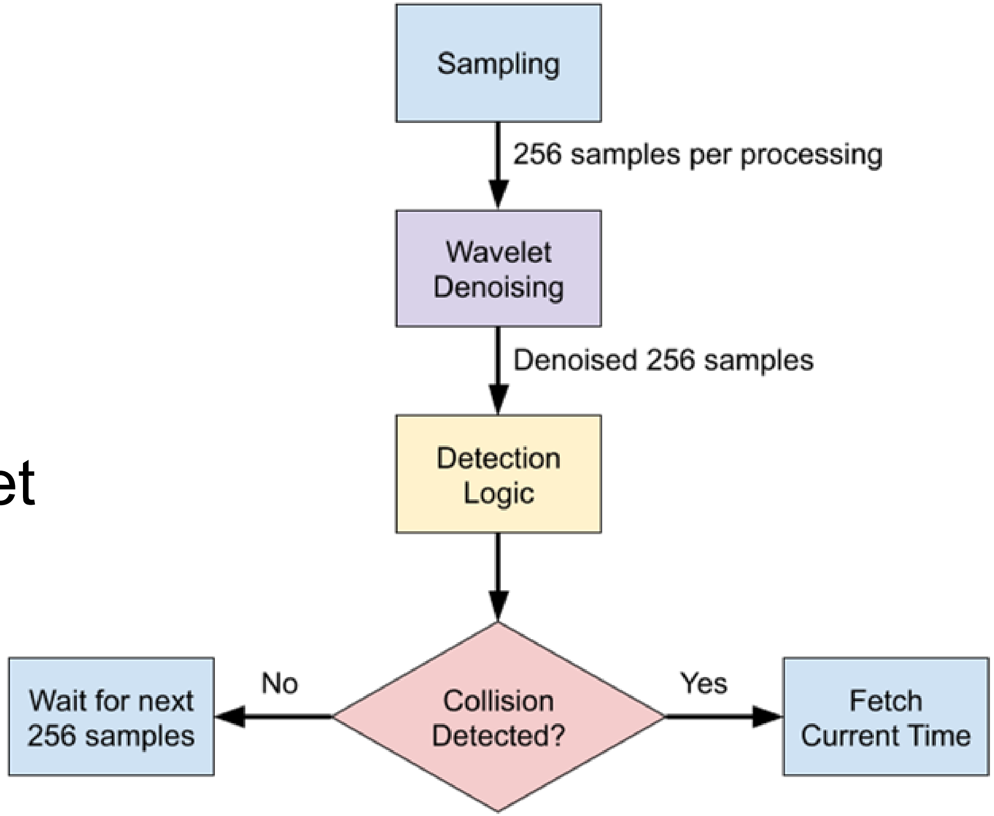
- Attached to the window using duct putty



Detection Algorithm

- Works with 256-sample frames of data

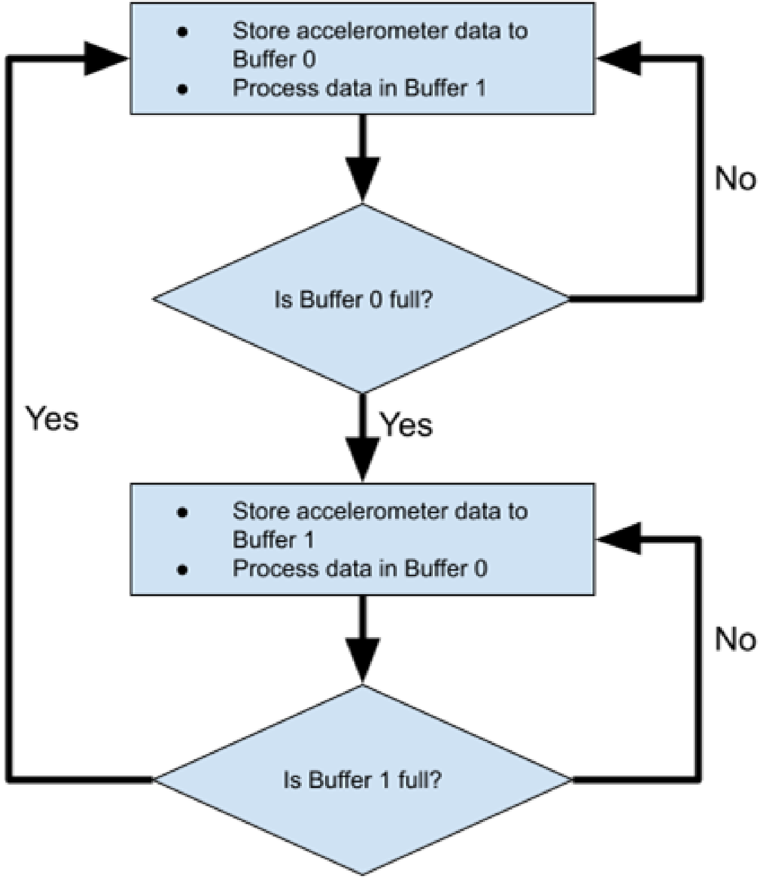
1. **Sampling:** read accelerometer at 400Hz
2. **Pre-processing:** denoise 256 samples using "wavelet denoising"
3. **Detection logic:** infer collision based on signal characteristics



Sampling

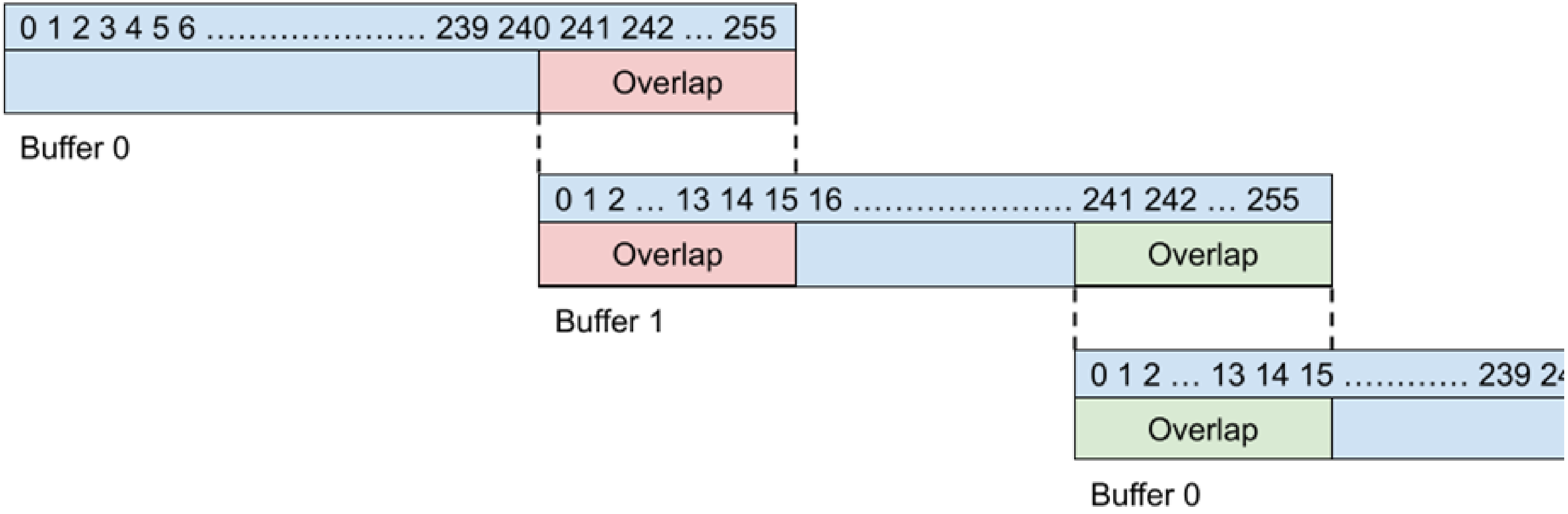
- Two 256-sample buffers to avoid overwriting

- One buffer stores new accelerometer readings
- Data in another buffer gets processed
- Role is switched once buffer is full



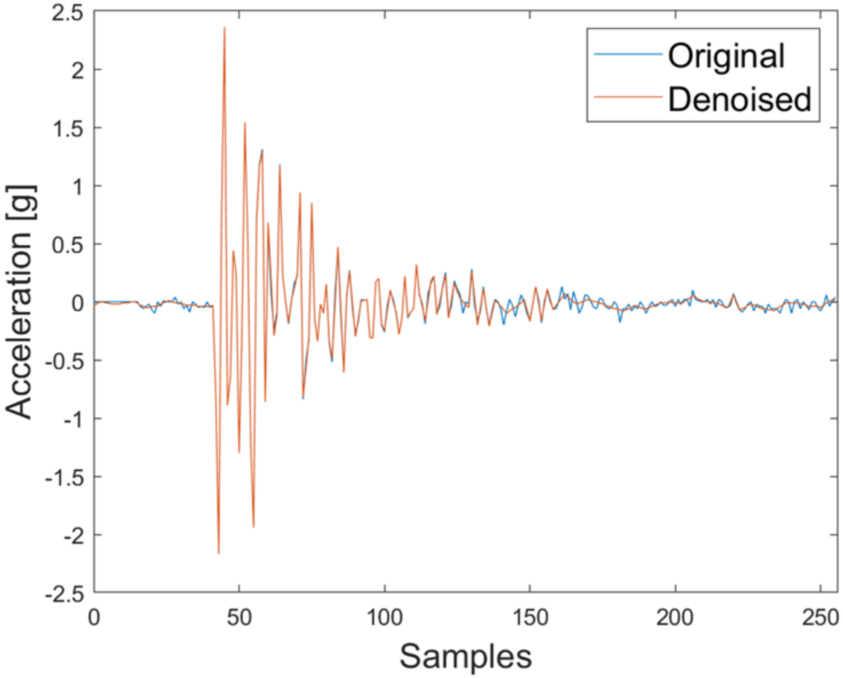
Sampling

- Denoising stage introduces filter delay
- 15 samples are overlapped to avoid loss of information



Wavelet Denoising

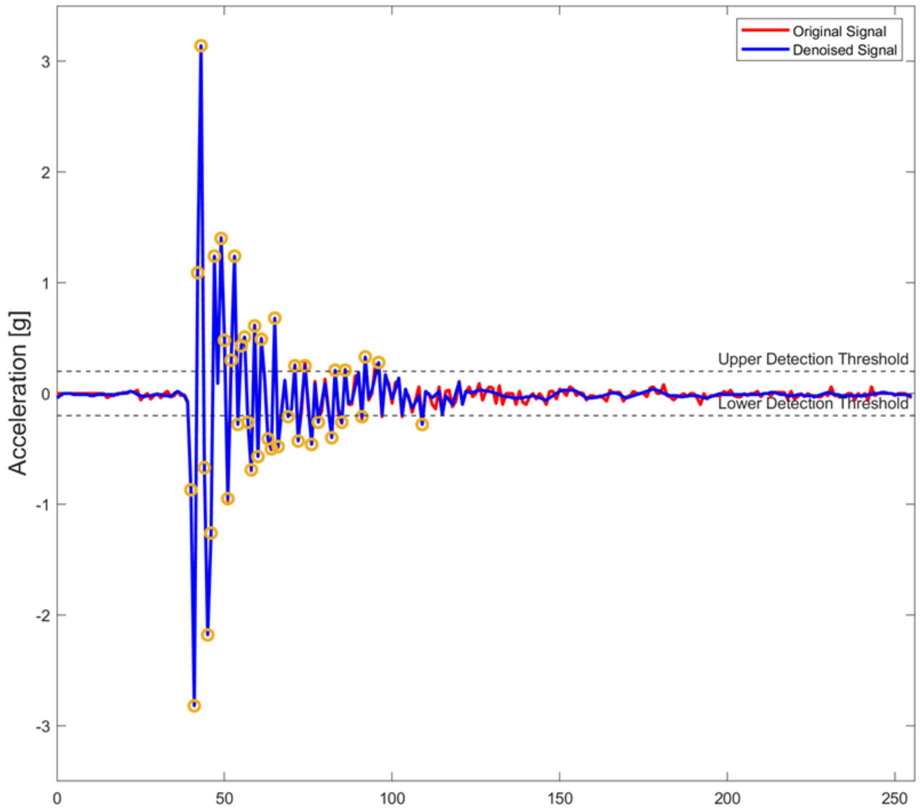
- **Goal:** Reduce noise without distorting the impact signal
- Common techniques like Moving Average Filter distort impact signal
- Filter bank implemented using CMSIS filter functions available for ARM Cortex-M microcontrollers
- **Execution time => ~32ms**



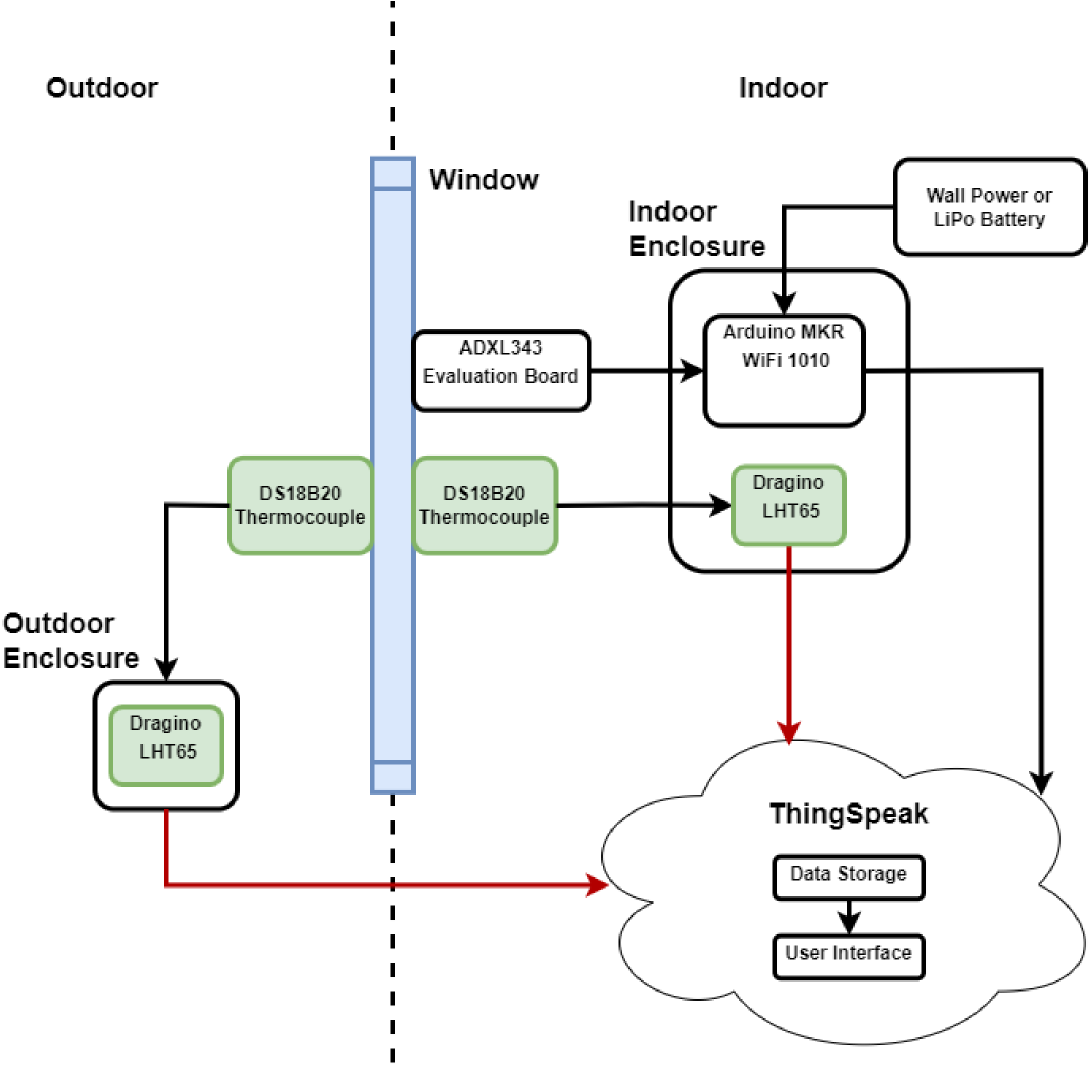
Detection Logic

Score-based algorithm:

1. Wait for a data point above detection threshold
2. Count number of data points above threshold (score)
3. Wait for multiple data points below threshold
4. Infer collision based on the score



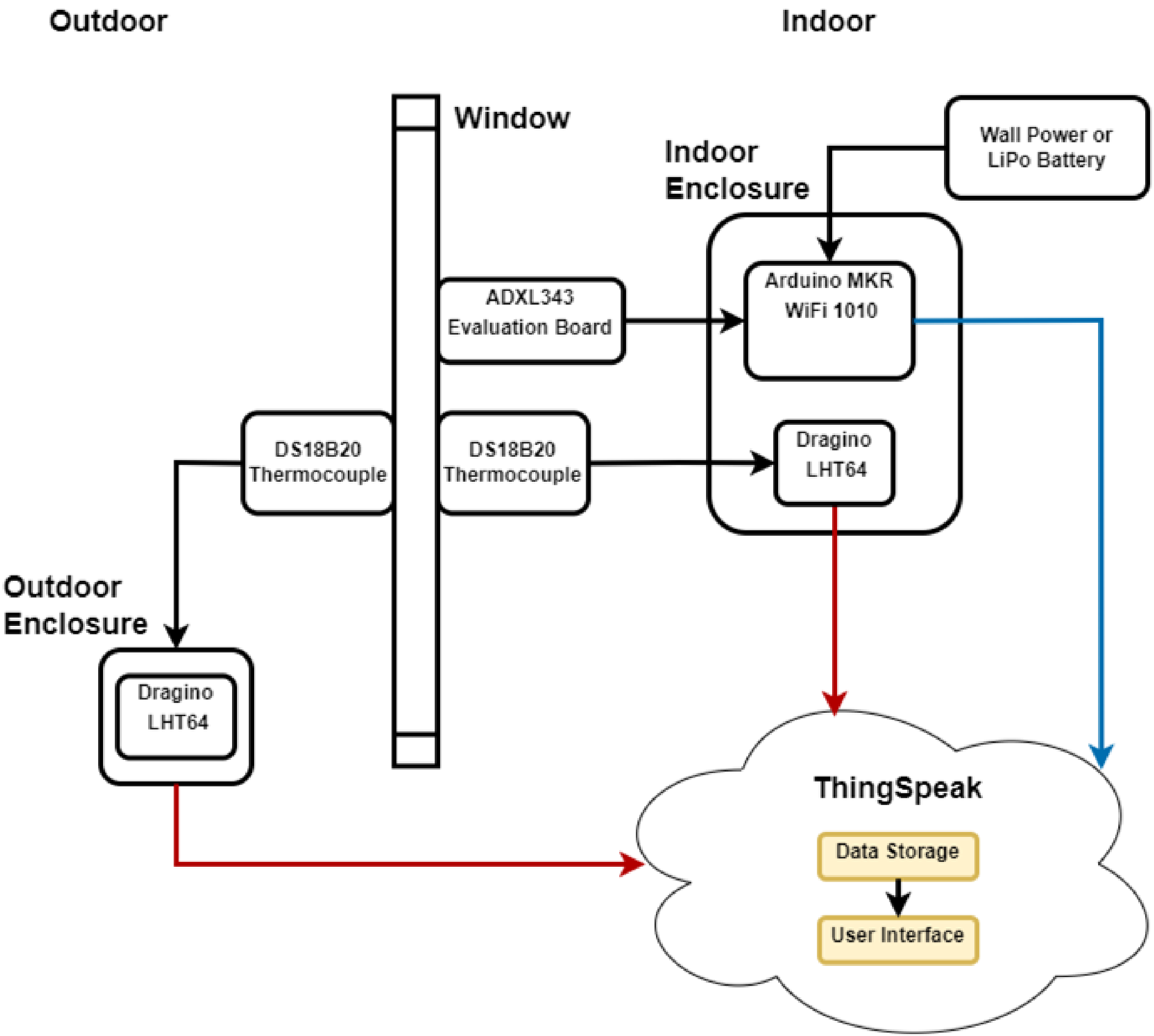
Heat Flow Rate Measurement



Heat Flow Rate System

- Dragino LHT 65 Temperature sensor is wireless, connect to LoRaWAN
- Thermocouple attached to window to measure surface temperature
- Data is used to calculate heat flow rate
- Uses LoRaWAN to send data to ThingSpeak

$$Heat\ Flow\ Rate = \frac{|Inside\ Surface\ Temperature - Outside\ Surface\ Tempearture|}{R\ value} \times Window\ Surface\ Area$$

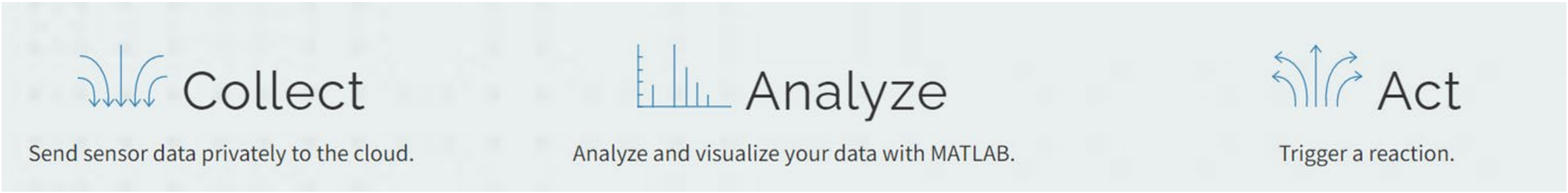


Data Storage & User Interface

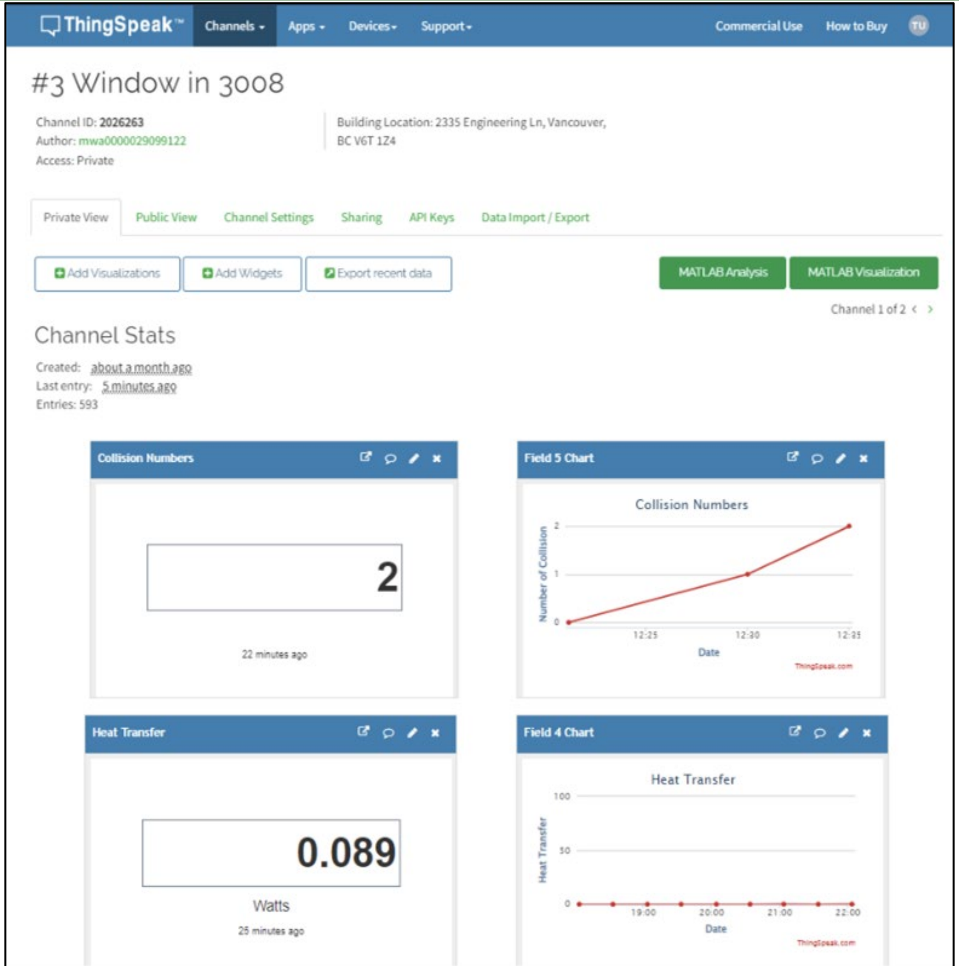
Data Storage & User Interface – ThingSpeak

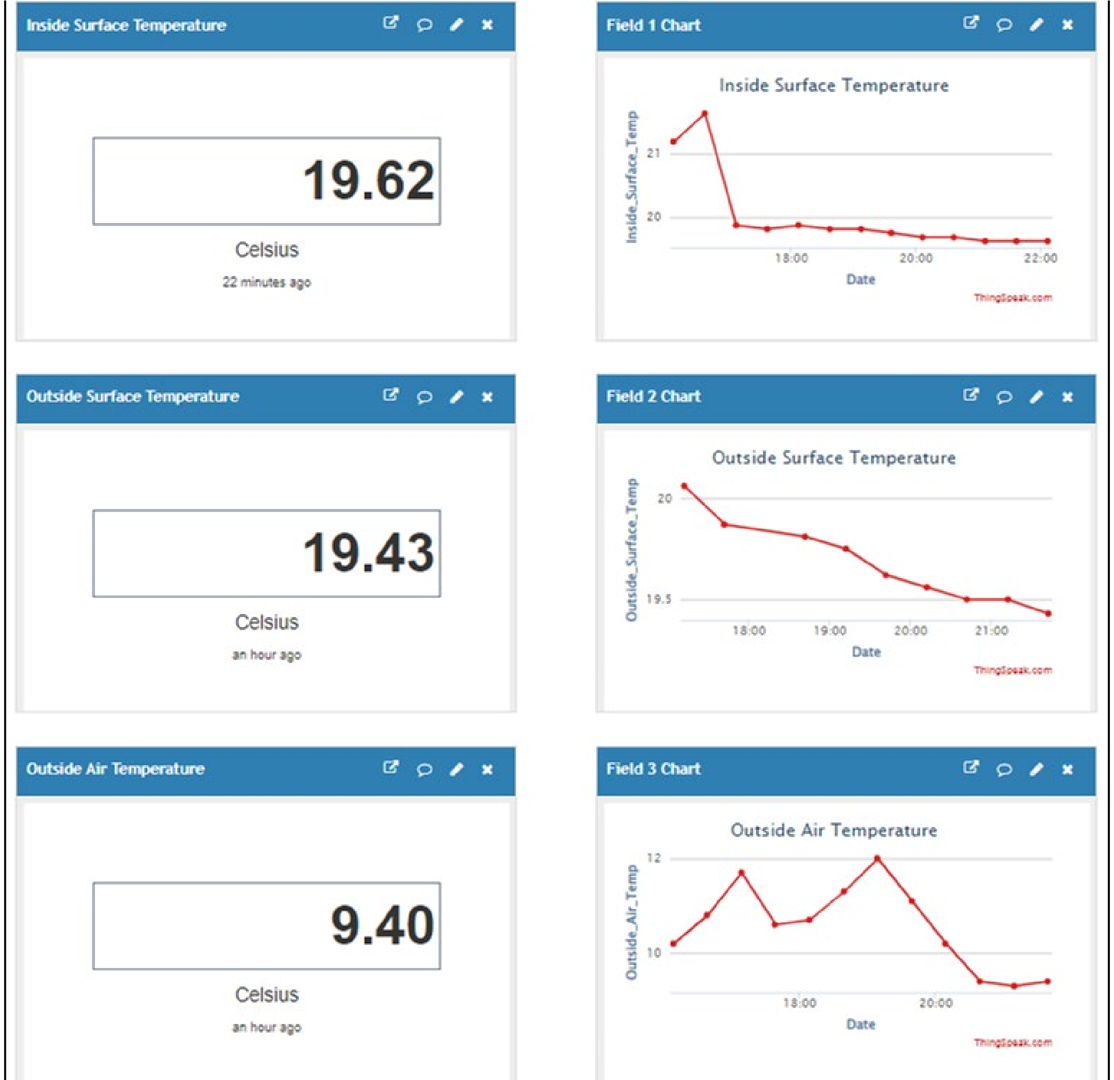
ThingSpeak™

- ~8,200 messages/day
- Up to 4 channels available
- MATLAB Analysis
- TimeControl
- Channel Dashboard

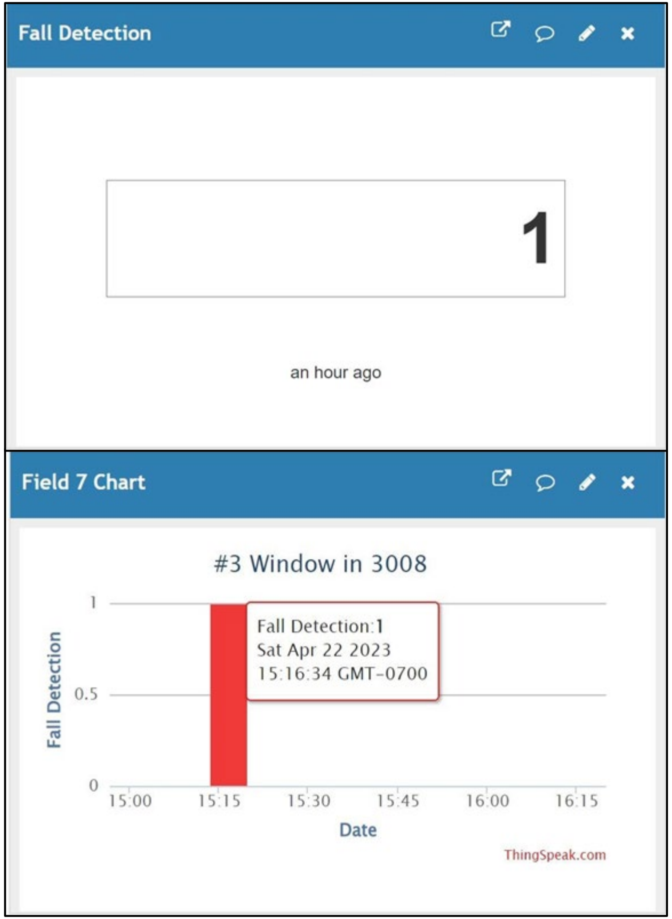


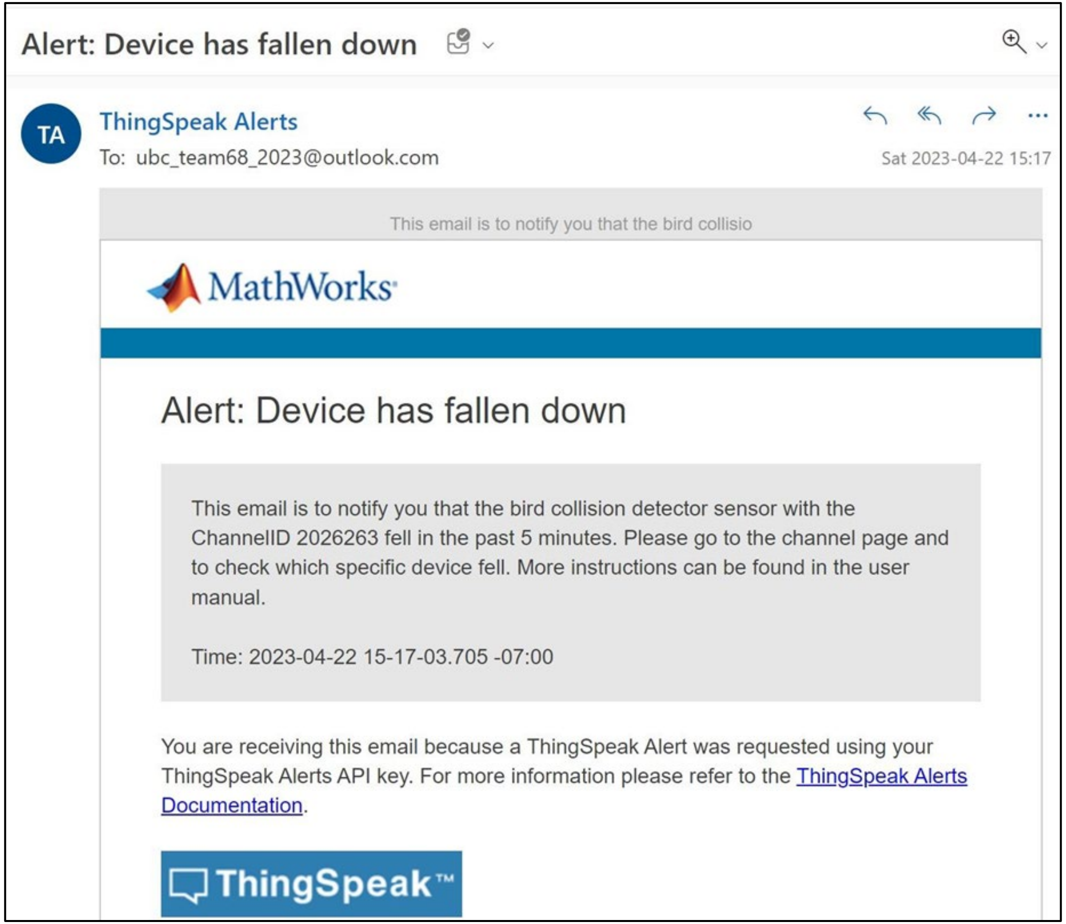
ThingSpeak – Dashboard




Inside Surface Temperature
19.62
Celsius
22 minutes ago
Field 1 Chart
Inside Surface Temperature
Inside_Surface_Temp
21
20
18:00
20:00
22:00
Date
ThingSpeak.com
Outside Surface Temperature
19.43
Celsius
an hour ago
Field 2 Chart
Outside Surface Temperature
Outside_Surface_Temp
20
19.5
18:00
19:00
20:00
21:00
Date
ThingSpeak.com
Outside Air Temperature
9.40
Celsius
an hour ago
Field 3 Chart
Outside Air Temperature
Outside_Air_Temp
12
10
18:00
20:00
Date
ThingSpeak.com

ThingSpeak – Email Notification for Fall Detection





Alert: Device has fallen down

TA ThingSpeak Alerts

To: ubc_team68_2023@outlook.com

Sat 2023-04-22 15:17

This email is to notify you that the bird collisio

MathWorks

Alert: Device has fallen down

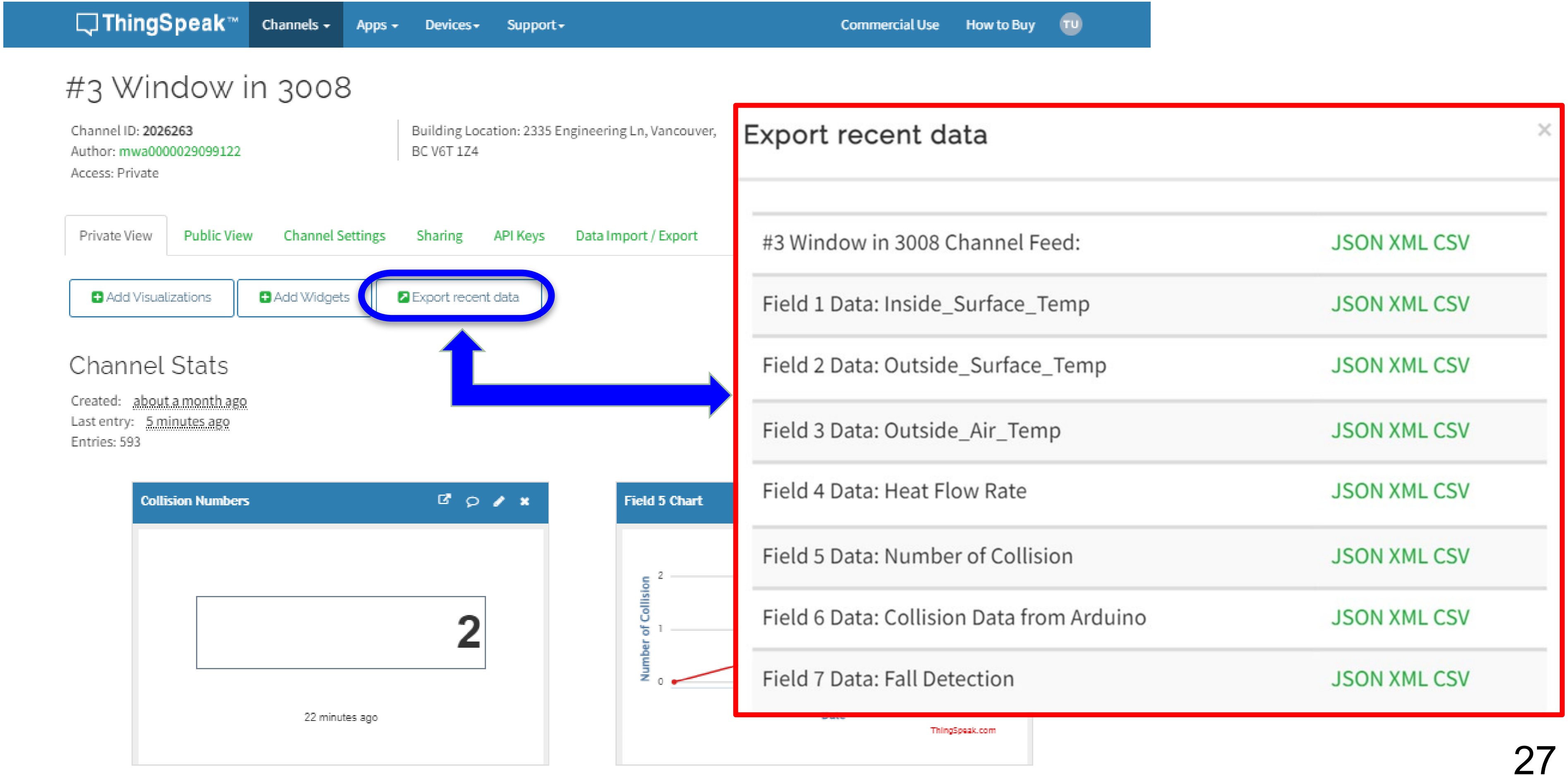
This email is to notify you that the bird collision detector sensor with the ChannelID 2026263 fell in the past 5 minutes. Please go to the channel page and to check which specific device fell. More instructions can be found in the user manual.

Time: 2023-04-22 15-17-03.705 -07:00

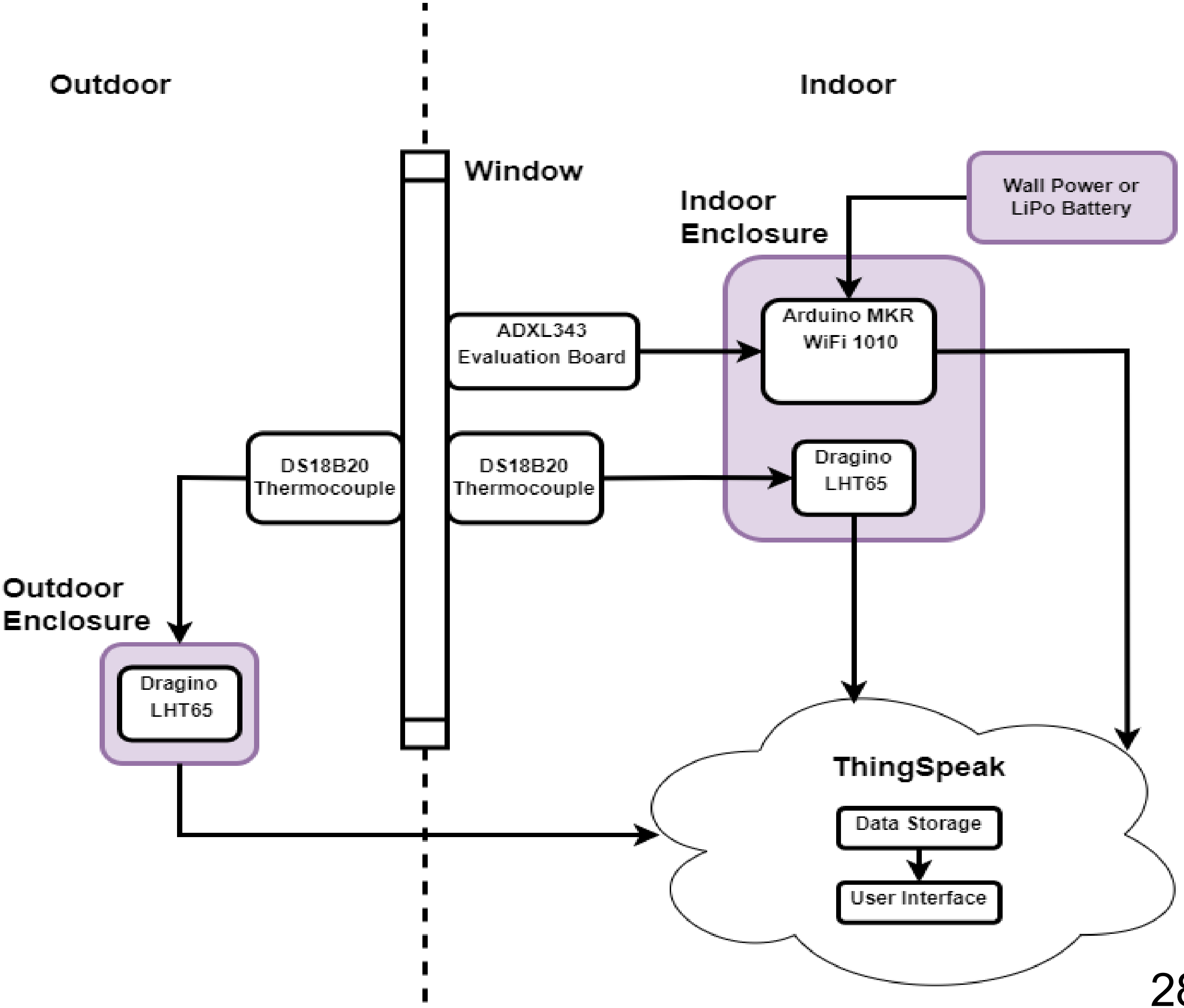
You are receiving this email because a ThingSpeak Alert was requested using your ThingSpeak Alerts API key. For more information please refer to the ThingSpeak Alerts Documentation.

ThingSpeak™

ThingSpeak – Data Exports

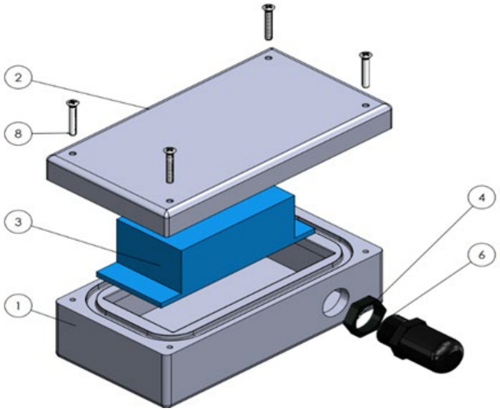


Hardware



Outdoor
Indoor
Window
Indoor Enclosure
Wall Power or LiPo Battery
Arduino MKR WiFi 1010
ADXL343 Evaluation Board
DS18B20 Thermocouple
DS18B20 Thermocouple
Dragino LHT65
Outdoor Enclosure
Dragino LHT65
ThingSpeak
Data Storage
User Interface

Hardware – Outside Case



- Printed in PETG for outdoor use
- Cable gland to pass temperature sensor through
- O-ring to create seal against ingress of water (ex. rain)

- Passed overnight rain test with no water getting in

Hardware - Inside Case

- Printed in FDM technology with PLA plastic material
- The case for the inside will contain the arduino MKR and the dragino sensor.

Hardware – Battery

- 2500mAh 3.7V Lithium-Polymer Battery
- Not dependent on power outlet
- Expands possible window locations to monitor
- 36 hour life

Demonstration

1. Collision detection
2. Fall detection

Limitations & Future Improvements

Limitations:

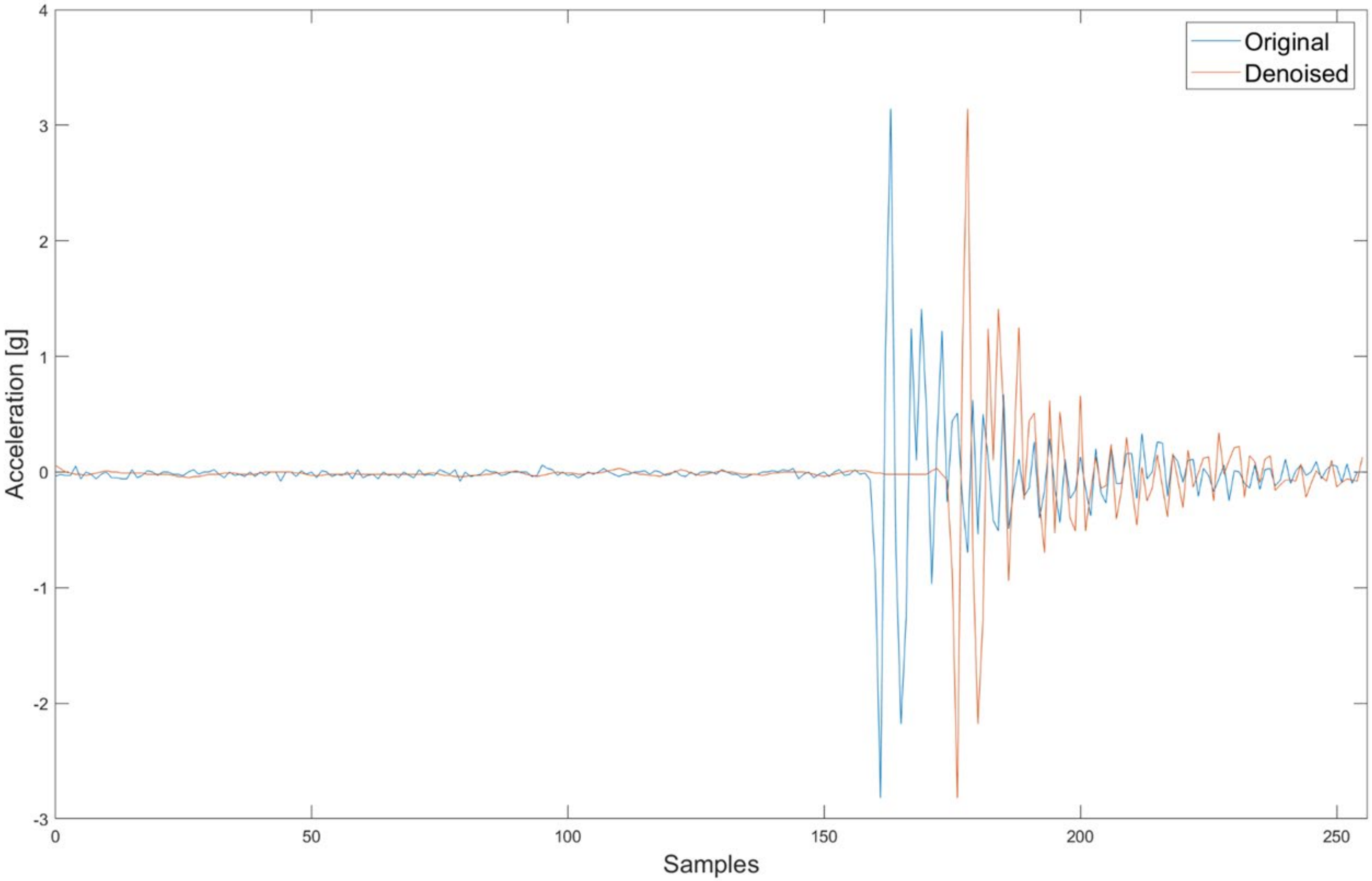
- Dependency to WiFi (high power consumption)
- Third-party dependency risk to UBC Visitor WiFi
- Potential false positives by detection algorithm (e.g., ball hitting the window)

Future Improvements:

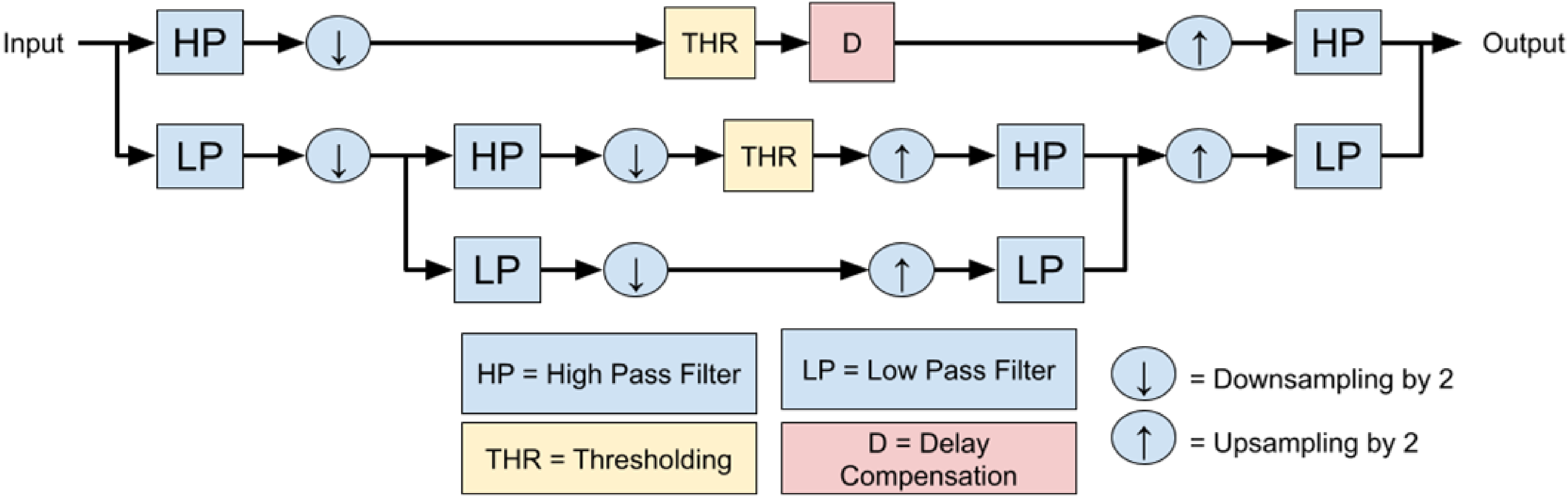
- More robust V&V to evaluate the detection algorithm performances
- Explore lower power mode on Arduino
- Higher capacity battery
- Multiple Accelerometers per Microcontroller

Thank you for listening!
Q&A

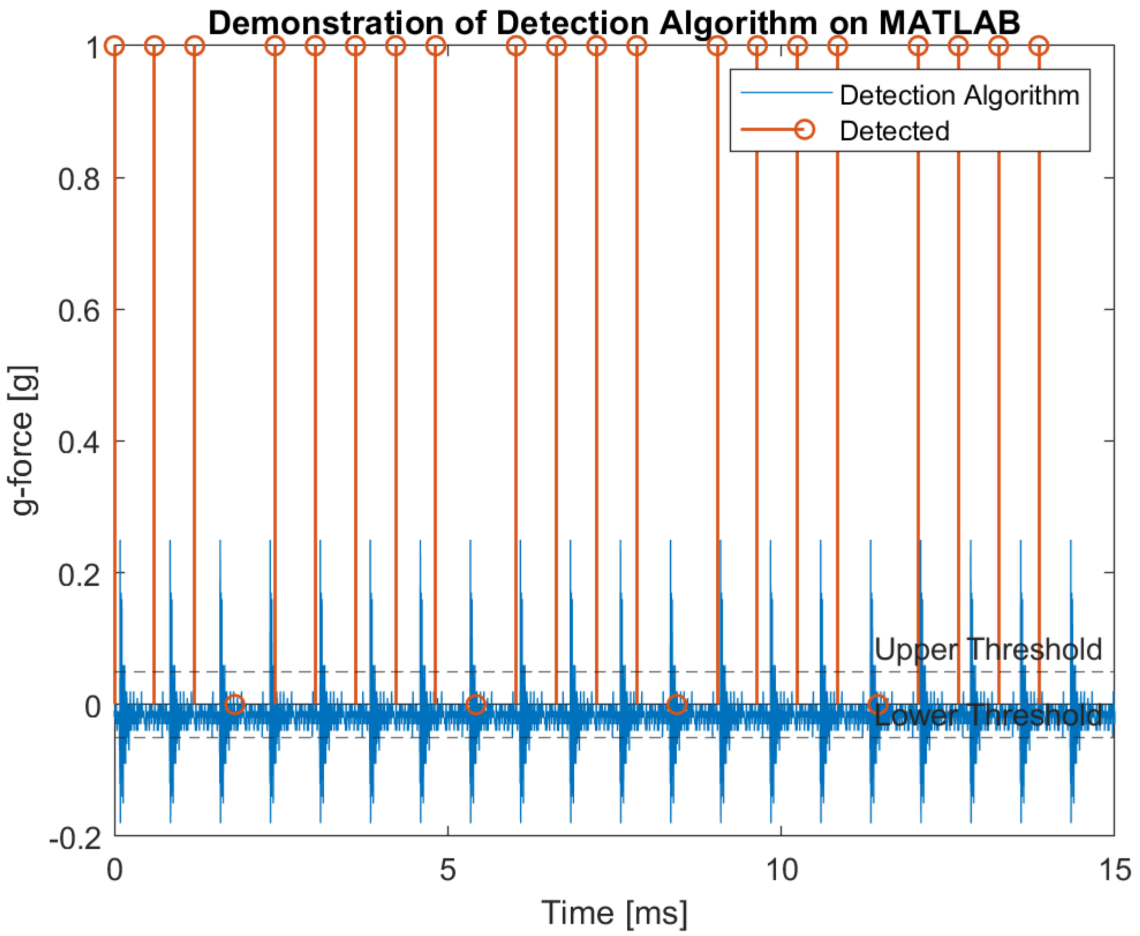
Supplementary – Border Effect



Supplementary - Wavelet Denoising Filter Bank



Supplementary – Detection Algorithm Output



Supplementary – Bill of Materials *Prices may vary

Item:	Qty:	Cost [CA$]:
Arduino MKR WiFi 1010	1	$52.12
ADXL343 Evaluation Board	1	$8.72
Ribbon Cable Assembly	1	$3.06
Pin Header	1	$1.33
LHT65 Temperature & Humidity Sensor	2	$120.98
O-ring	1	$1.37
Cable Gland	1	$2.43
Locknut	1	$0.66
Micro USB Cable	1	Cost depends on the retailer
Lithium Ion Polymer Battery - 3.7v 2500mAh	1	$20.19
3D Printed Enclosures (x4)	1	$269.71
	Total:	**$480.57**

Supplementary – *Functional Requirement*

F1: The system must detect collisions with windows and infer whether it is a false-positive collision based on calculated force and frequency metrics of the collision.
F2: The system must record the cumulative number of birds strikes along with the time of day, season and date at which the strikes occurred.
F3: The system must record the latitude and longitude of the building that window placed
F4: The system must record the data packet for the fall detection.
F5: The system must record the surface temperature of the window inside and outside.
F6: The system must record the air temperature outside of the building.
F7: The device must have the ability to secure itself to the window and no part of the device can be left unattached to, or separated from, the window under normal operation and avoid dropping in adverse weather conditions.
F8: The user interface must graphically display stored data.
F9: The user interface must export historical data from the database to the user's computer as a file.
F10: The system must alert the user in the case of the accelerometer falling from its attached position on the selected window.

Supplementary – *Non-Functional Requirement*

NF1: The accuracy of bird strike detection must be higher than 95%. (Using simulated bird strikes.)

NF2: The calculation error of the heat flux should be within ±0.001 Watts/m^2

NF3: Th measurement error of the window's surface temperature and air temperature must be within ±2°C.

NF4: The data recording must be fully functional for a minimum of 30 days before requiring maintenance

NF5: The outside part of the device must be waterproof to avoid water ingress in the event of rain. Reach the IP standard IP23.

NF6: The device must not have color or shape features that attract bird species and/or other animals.

NF7: The database must have sufficient storage size to store at least 10 bird impact events per day.

NF8: The temperature sensors should measure the temperature per hour.

Supplementary – *Constraints*

C1: The whole project cost (including hardware, software, and auxiliary supplies) should be under $180.

C2: Materials chosen for the device must withstand the operating temperature of between -20°C ~ 40°C. (±6°C based on maximum and minimum temperatures of the past year)

C3: The device must operate on a wireless network.

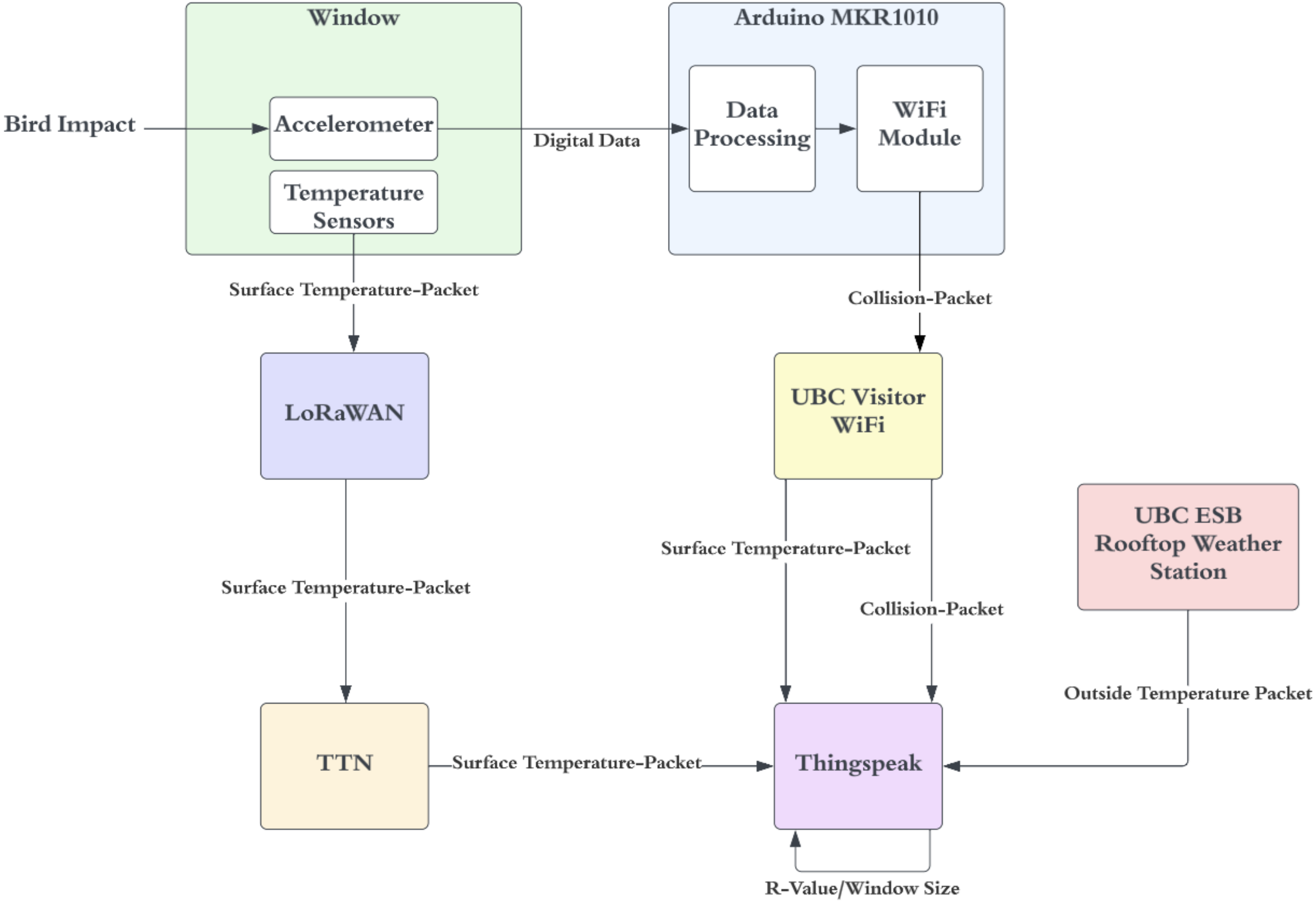
C4: The device should support both battery charging and wall outlet 120VAC charging methods.

C5: The external and internal hardware design should be within 15cm*6cm*6cm dimensions to achieve a compact design.

C6: The weight of the entire hardware design (Both external and internal) should be within the 200 g.

C7: The system must function on a window with dimension 80cm * 50cm. Bird collision detection model must work on typical windows on the UBC campus.

System Data Pathway



Supplementary – *Power Consumption Calculations*

Calculate the time spent in each task per day:

- ***Task 1 and Task 2:*** *24 hours - (288 x 10 seconds for Task 3) - (1 minute for Task 4) = 23.183 hours*
- ***Task 3:*** *288 x 10 seconds = 2880 seconds or 0.8 hours*
- ***Task 4:*** *60 seconds = 0.01666 hours*

Next, calculate the weighted average power consumption for each task:

- ***Task 1 and Task 2:*** *(70 mA) x (23.183 hours / 24 hours) ≈ 67.618 mA*
- ***Task 3:*** *(70 mA) x (0.8 hours / 24 hours) ≈ 2.3 mA*
- ***Task 4:*** *(150 mA) x (0.01666 hours / 24 hours) ≈ 0.1042mA*
- ***System wide:*** *67.618 mA + 2.3 mA + 0.1042 mA ≈ 70.052 mA*

Finally, calculate the battery life:

- ***Battery life (h) = Battery capacity (mAh) / Average power consumption (mA)***

  = 2500 mAh / 70.052 mA ≈ 35.68 h

Supplementary – Minimum Data Storage Calculation Process

$$Number\ of\ outside\ surface\ tempearture\ data\ entries\ per\ day = \left(\frac{60\ mins}{30\ mins} * 24hrs\right) = 48$$

$$Number\ of\ inside\ surface\ tempearture\ data\ entries\ per\ day = \left(\frac{60\ mins}{30\ mins} * 24hrs\right) = 48$$

$$Number\ of\ outside\ air\ tempearture\ data\ entries\ per\ day = \left(\frac{60\ mins}{30\ mins} * 24hrs\right) = 48$$

$$Number\ of\ heat\ flow\ rate\ data\ entries\ per\ day = \left(\frac{60\ mins}{30\ mins} * 24hrs\right) = 48$$

$$Number\ of\ collision\ detection\ data\ entries\ per\ day$$
$$= 2\ \ collision\ data\ segments\ sent\ from\ the\ microcontroller$$
$$+ 1\ analyzed\ bird\ impact\ counts\ data\ segments\ being\ stored\ in\ the\ database\ \ = 3$$

$$Number\ of\ fall\ detection\ data\ entries\ per\ day = 1$$

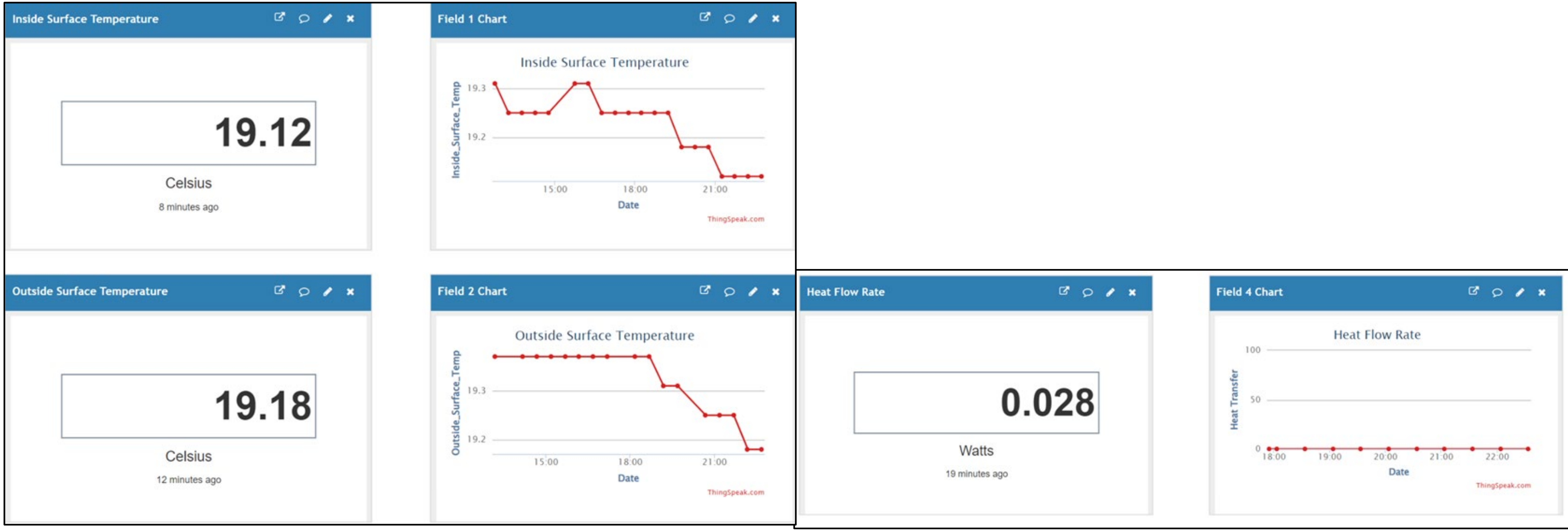
$$Number\ of\ total\ data\ entries\ per\ day = 48 * 4 + 3 + 1 = 196$$

$$Total\ storage\ size\ required\ each\ day = \left(\frac{3790\ bytes\ within\ 12\ hrs}{102\ data\ entries\ within\ 12\ hrs} * 196\ data\ entries\ per\ day\right) \approx 7283\ bytes\ per\ day$$

$$7283\ bytes\ per\ day = \frac{7283}{1024}\ kilobytes\ per\ day = 7.1KB\ per\ day$$

$$Total\ storage\ size\ required\ per\ \ 30\ day\ period = \ 7.1KB\ per\ day * 30 = 213KB$$

Supplementary – Heat Flow Rate System



Supplementary – Arduino Monitor

```
18:35:51.693 -> .....Task 1: 256 samples collected
18:35:52.294 -> Task 2: Detection algorithm begin
18:35:52.334 -> detected
18:35:52.383 -> 183551
18:35:52.383 -> .....Task 1: 256 samples collected
18:35:52.955 -> Task 2: Detection algorithm begin
18:35:53.003 -> detected
18:35:53.003 -> 183552
18:35:53.003 -> .....Task 1: 256 samples collected
18:35:53.625 -> Task 2: Detection algorithm begin
18:35:53.673 -> .....Task 1: 256 samples collected
18:35:54.243 -> Task 2: Detection algorithm begin
```

```
18:37:49.222 -> .....Task 1: 256 samples collected
18:37:49.827 -> Task 2: Detection algorithm begin
18:37:49.875 -> .....Task 4: Sending data to ThingSpeak start
18:37:50.390 -> 2
18:37:52.860 -> start1
18:37:52.860 -> Loop0
18:37:53.238 -> Updating channel. HTTP code 200
18:38:13.248 -> Loop1
18:38:13.627 -> Updating channel. HTTP code 200
18:38:33.634 -> Loop2
18:38:34.120 -> Updating channel. HTTP code 200
18:38:34.120 -> Task 4: Sending data to ThingSpeak End
18:38:34.120 -> Task 2: Detection algorithm begin
18:38:34.120 -> .Task 1: 256 samples collected
18:38:34.168 -> Task 2: Detection algorithm begin
```

```
18:28:43.503 -> Please upgrade the firmware
18:28:43.503 -> Attempting to connect to WPA SSID: ubcvisitor
18:28:50.683 -> SSID: ubcvisitor
18:28:50.683 -> BSSID: CC:DB:93:44:A7:C5
18:28:50.683 -> signal strength (RSSI):-78
18:28:50.683 -> Encryption Type:7
18:28:50.683 ->
18:28:50.683 -> IP Address: 137.82.226.173
18:28:50.683 -> 137.82.226.173
18:28:50.683 -> MAC address: 24:62:AB:B9:33:94
18:28:50.732 -> 28
18:28:50.732 -> Task 3: Fall Detection Test
18:28:50.732 -> Accelerometer Y value:1.06
18:28:50.732 -> Task 3: Fall Detection Test END
18:28:50.732 -> Task 4: Sending data to ThingSpeak start
18:28:50.732 -> 0
18:28:53.238 -> Task 4: Sending data to ThingSpeak End
18:28:53.238 -> Task 2: Detection algorithm begin
18:28:53.238 -> ......Task 1: 256 samples collected
18:28:53.865 -> Task 2: Detection algorithm begin
18:28:53.993 -> .....Task 1: 256 samples collected
18:28:54.515 -> Task 2: Detection algorithm begin
```

```
19:14:01.870 -> Task 3: Fall Detection Test
19:14:01.870 -> Accelerometer Y value:-0.15
19:14:01.870 -> Fall1
19:14:01.870 -> Task 3: Fall Detection Test END
19:14:01.870 -> Task 4: Sending data to ThingSpeak start
19:14:01.870 -> 0
19:14:04.413 -> Task 4: Sending data to ThingSpeak End
19:14:04.413 -> Task 2: Detection algorithm begin
19:14:04.413 -> ......Task 1: 256 samples collected
19:14:05.050 -> Task 2: Detection algorithm begin
19:14:05.134 -> .....Task 1: 256 samples collected
19:14:05.648 -> Task 2: Detection algorithm begin
```

```
19:19:00.484 -> .....Task 1: 256 samples collected
19:19:01.085 -> Task 2: Detection algorithm begin
19:19:01.134 -> .....Task 3: Fall Detection Test
19:19:01.700 ->
19:19:01.700 -> Still not Press Reset Button Yet
19:19:01.700 -> FALL EMAIL
19:19:02.129 -> Updating channel. HTTP code 200
19:19:02.129 -> 0Still not Press Reset Button Yet
19:19:02.129 -> 0Still not Press Reset Button Yet
19:19:02.129 -> 0Still not Press Reset Button Yet
19:19:02.129 -> 0Still not Press Reset Button Yet
```

```
19:19:07.199 -> 0Still not Press Reset Button Yet
19:19:07.199 -> 0Still not Press Reset Button Yet
19:19:07.199 -> 1Reset Press
19:19:07.199 -> Resume Task B and C First
19:19:07.199 -> Task 2: Detection algorithm begin
19:19:07.199 -> Task 4: Sending data to ThingSpeak start
19:19:07.199 -> 0
19:19:09.686 -> Task 4: Sending data to ThingSpeak End
19:19:09.734 -> Task 2: Detection algorithm begin
19:19:09.734 -> .Task 1: 256 samples collected
19:19:09.734 -> Task 2: Detection algorithm begin
19:19:09.815 -> .....Task 1: 256 samples collected
19:19:10.355 -> Task 2: Detection algorithm begin
```