



**DesignNews**

Getting Started With LoRaWAN and Sensor IoT

# DAY 4: Creating a LoRaWAN

Sponsored by

**DigiKey**



## Webinar Logistics

- Turn on your system sound to hear the streaming presentation.
- If you have technical problems, click “Help” or submit a question asking for assistance.
- Participate in ‘Attendee Chat’ by maximizing the chat widget in your dock.



## Dr. Don Wilcher

Visit 'Lecturer Profile' in your console for more details.

LinkedIn Page:

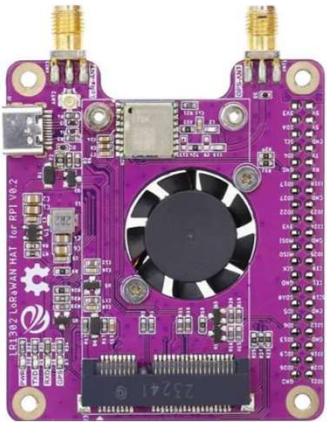
<https://www.linkedin.com/in/dr-don-wilcher-ed-d-mseit-ee-ceta-2735151/>

Patreon Page:

<https://www.patreon.com/c/DrDon683>

## Course Kit and Materials

**Elecrow LR1302 LoRaWAN Pi Hat**



**Raspberry Pi 4 Single Board Computer**



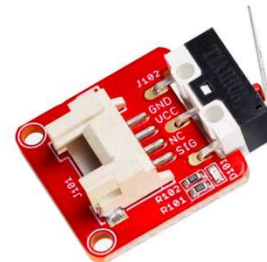
**USB-C Cable**



**Elecrow Crowtail LED 2.0**



**Elecrow Crowtail Collision Sensor**



**LR1302 Gateway Module**



**Elecrow LR1262 Node Board**



## Research Perspective

- “LoRaWAN is one of the promising technologies of the Low Power Wide Area Network (LPWAN) that provides communication with low power, low cost, long-range, and low data rate (Rahman et al., 2020).”

## Agenda:

- What are Crowtails?
- Establishing a Single Channel LoRaWAN
- Modifying the LoRa 08H Code
- Lab: Build and Test a LoRaWAN

## What are Crowtails?

- Crowtails are a type of modular electronic component.
- A line of modules from Elecrow and other brands, designed for Arduino and other microcontroller platforms.
- They are similar in concept to Grove modules by Seeed Studio or Qwiic modules by SparkFun.
- They offer a convenient way to add features like sensors, LEDs, buttons, and more to projects, simplifying wiring and making them easier for beginners to use.



## What are Crowtails? ...



### Plug-and-Play System:

- Use standard 4-pin connectors
- Cables to connect to a Crowtail Base Shield without soldering or complex wiring.

### Compatibility:

- Designed to work with Arduino, micro:bit, and other microcontrollers (like Raspberry Pi with adapters).
- Crowtail Base Shields are available for these platforms.

## What are Crowtails? ...

### Wide Range of Modules:

#### Input modules:

- a) buttons
- b) potentiometers
- c) sensors
  - i. temperature
  - ii. Light
  - iii. Sound

#### Output modules:

- a) LEDs
- b) buzzers
- c) displays,
- d) relays
- e) etc.

#### Communication modules:

- a) Bluetooth
- b) Wi-Fi
- c) UART
- d) I2C devices

#### Actuators:

- a) Servos
- b) motors
- c) etc.



## Question 1

**Which concept is the Crowtail similar to?**

- a) Groove modules**
- b) Quick modules**
- c) Grove or Quic modules**
- d) none of the above**



# What are Crowtails? ...



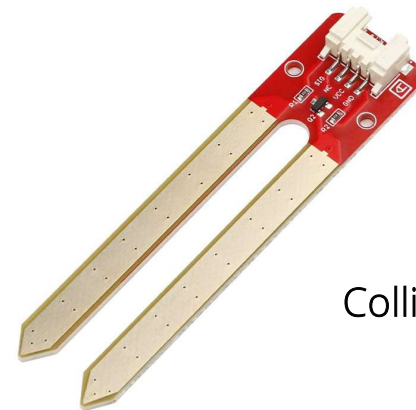
Sound Sensor



Crowtail Examples  
LED



Moisture Sensor



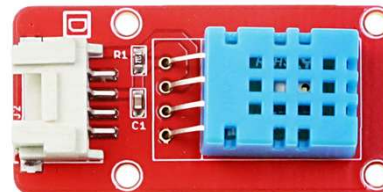
Servo



Collision Sensor



Temperature-Humidity  
Sensor



# What are Crowtails? ...

## Exploring the Collision Sensor

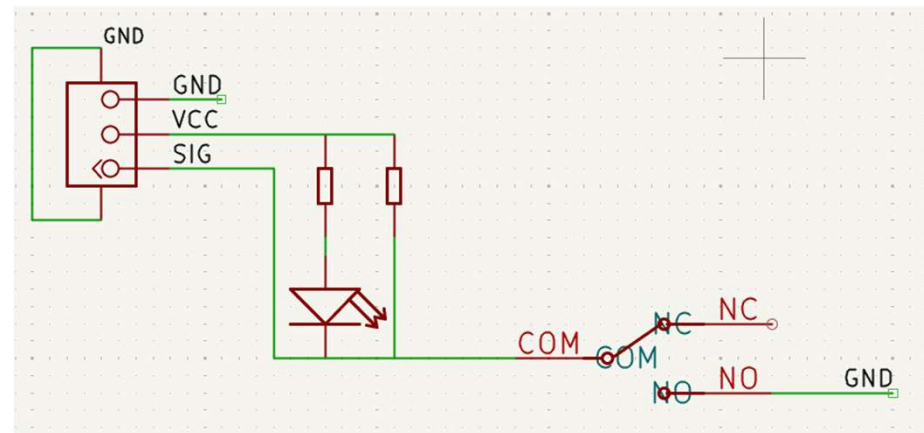


Collision Sensor



### Collision Sensor Wiki

<https://www.elecrow.com/wiki/crowtail--collision-sensor.html>



Collision Sensor Electronic Circuit Schematic Diagram

## What are Crowtails? ...

### Exploring the Collision Sensor: Arduino Test Code



```
1 |
2 | const int sensorPin = 2;
3 | const int ledPin = 5;      // the number of the LED pin
4 |
5 | // variables will change:
6 | int sensorState = 0;
7 |
8 | void setup() {
9 |     // initialize the LED pin as an output:
10 |    pinMode(ledPin, OUTPUT);
11 |    pinMode(sensorPin, INPUT);
12 | }
```

```
13 |
14 | void loop() {
15 |     // read the state of the pushbutton value:
16 |     sensorState = digitalRead(sensorPin);
17 |     if (sensorState == HIGH) {
18 |         // turn LED on:
19 |         digitalWrite(ledPin, LOW);
20 |     }
21 |     else {
22 |         // turn LED off:
23 |         digitalWrite(ledPin, HIGH);
24 |     }
25 | }
```

## Question 2

**In reviewing slide 13, what digital device is being detected by the Arduino Test Code?**

- a) An Encoder**
- b) 555 Timer One-Shot Circuit**
- c) Pulsimeter**
- d) Digital Inverting Switch**



## Establishing a Single Channel LoRaWAN



To create a single-channel LoRaWAN network using:

- a) Elecrow LR1302 LoRaWAN Gateway Module with Pi Hat
- b) Elecrow LR1262 Node

The gateway and node (mote) will need to be configured appropriately so they can communicate over a single frequency and spreading factor, which is common for:

- a) testing
- b) prototyping
- c) educational purposes.

## Establishing a Single Channel LoRaWAN



### What is a Single-Channel LoRaWAN Network?

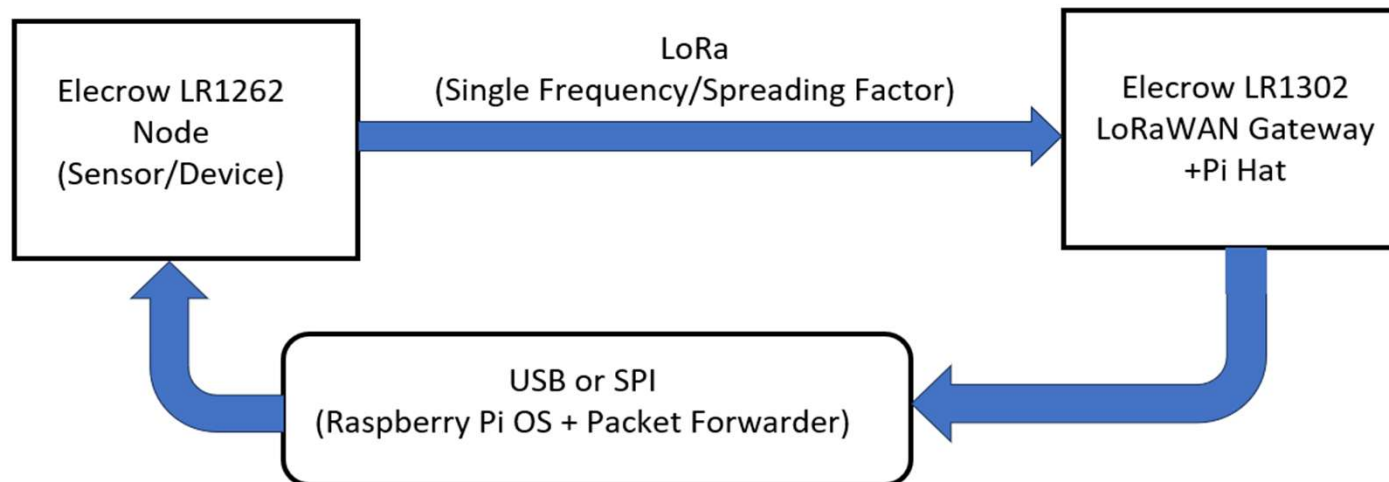
- A single-channel gateway listens on:
  - a) one frequency
  - b) spreading factor
- Unlike full gateways, single-channel communication systems do not support multiple channels.
- While not LoRaWAN-compliant for production, single-channel setups are suitable for learning and development.

**Note:** The following development and setup steps were generated by ChatGPT but are aligned with Elecrow's Wiki Case 3:

[https://www.elecrow.com/wiki/Case3\\_Single-channel\\_Application\\_Routines\\_for\\_Raspberry\\_Pi\\_Gateways\\_and\\_Nodes.html#frequency-selection](https://www.elecrow.com/wiki/Case3_Single-channel_Application_Routines_for_Raspberry_Pi_Gateways_and_Nodes.html#frequency-selection)

## Establishing a Single Channel LoRaWAN

### What is a Single-Channel LoRaWAN Network?



Single Channel LoRaWAN Network Architecture Diagram

## Question 3

**Which statement is incorrect about a single-channel gateway?**

- a) A single-channel gateway listens on one frequency and two spreading factors.**
- b) A single-channel gateway listens on two frequencies and 2 spreading factors.**
- c) A single-channel gateway listens on one frequency and one spreading factor.**
- d) none of the above**



## Establishing a Single Channel LoRaWAN...



### What You Need:

- Raspberry Pi (connected to the LR1302 Pi Hat Gateway Module)
- LR1302 LoRa Gateway Module with SX1302 chip
- LR1262 Node (based on Semtech LR1262)
- Power supply for both devices
- USB or UART interface for programming the Node
- LoRaWAN stack software (like `packet forwarder`, `chirpstack`, or `basicstation`) on the gateway
- LoRaMAC-node or Arduino-compatible firmware for the Node

## Establishing a Single Channel LoRaWAN...



### Steps to Set Up a Single-Channel Network

#### Step 1: Setup the LR1302 Gateway Module with Pi Hat

1. Install Raspberry Pi OS on your Pi.
2. Connect the LR1302 Pi Hat to the GPIO header.
3. Enable SPI interface:

```
sudo raspi-config
```

Go to *Interfacing Options* > *SPI* > *Enable*.

## Establishing a Single Channel LoRaWAN...



### 4. Download and install LoRa gateway software:

For single-channel operation, you can use a modified `packet_forwarder` or ChirpStack Gateway OS Lite.

Example (install Semtech packet forwarder for testing):

```
git clone https://github.com/Lora-net/packet_forwarder.git
cd packet_forwarder
make
```

### 5. Configure the Gateway for:

- **Single frequency** (e.g., 868.1 MHz or 915.0 MHz depending on your region)
- **Spreading Factor** (e.g., SF7)
- **LoRa mode** (LoRa-only)

## Establishing a Single Channel LoRaWAN...



### ✓ Step 2: Configure the LR1262 Node

#### 1. Install LoRa firmware on the Node:

- Use the **Arduino IDE** or **STM32CubeIDE** if the node is STM32-based.
- Choose a **LoRa P2P** or **LoRaWAN** firmware that allows setting frequency and SF.

#### 2. Set the LoRa parameters to match the gateway:

- Frequency: Same as gateway (e.g., 915.0 MHz)
- Spreading Factor: Same (e.g., SF7)
- Bandwidth: e.g., 125 kHz
- Coding Rate: 4/5

#### 3. Example Arduino-style snippet:

```
LoRa.begin(915E6); // US frequency
LoRa.setSpreadingFactor(7);
LoRa.setSignalBandwidth(125E3);
LoRa.setCodingRate4(5);
```

## Establishing a Single Channel LoRaWAN...



### ✓ Step 3: Test Communication

- On the **Node**, send a test payload:

```
LoRa.beginPacket();  
LoRa.print("Hello Gateway!");  
LoRa.endPacket();
```

- On the **Gateway**, use a log or UART serial terminal to check if the message is received.

## Establishing a Single Channel LoRaWAN...



### Tech Notes

- Single-channel gateways can't support full LoRaWAN node class functionality (no downlinks, no ADR).
- Best for **controlled, small-scale tests** (range tests, payload tests).
- Use **ChirpStack** or **TTN Gateway Bridge** only if you're simulating full LoRaWAN (not needed for simple point-to-point).

## Modifying the LoRa 08H Code...



- The original LoRa 08H Code allows for the testing and evaluating the LR1262 Node (mote).
- AT Commands are sent to the LR1262 for sending configuration settings and receiving information like:
  - a) joining a LoRaWAN.
  - b) the LR1262 Node functional parameters.
    - i. battery status (AT+BAT=?)
    - ii. version (AT+VER=?)
    - iii. Data Rate (DR) [AT+DR=?]

## Modifying the LoRa 08H Code...



LoRa 08H Code

```
1 void setup() {  
2     // put your setup code here, to run once:  
3     Serial.begin(9600);  
4     Serial1.setRX(1);  
5     Serial1.setTX(0);  
6     Serial1.begin(9600);  
7 }  
8  
9 void loop() {  
10     if (Serial.available())  
11     {  
12         Serial1.write(Serial.read());  
13     }  
14  
15     if (Serial1.available())  
16     {  
17         Serial.write(Serial1.read());  
18     }  
19 }
```

## Modifying the LoRa 08H Code...



The Modified LoRa 08H Code's purpose.

To illustrate the LoRaWAN functionality, the LoRa 08H code can be modified with the following features.

- a) Demonstrate data flow from the LR1262 Node to the Gateway (LR1302 LoRaWAN+Pi Hat).
- b) The use of a Collision Sensor Crowtail to represent data being transmitted from a remote node (mote).
- c) To view the transmitted data on an external monitor connected to the Raspberry Pi.

## Modifying the LoRa 08H Code...



These AT Parameters can be obtained using the LoRa 08H code!

```
1 #define LED_PIN 3
2 #define BUTTON_PIN 2
3 #define BLINK_DURATION 100 // milliseconds
4
5 const char* atCommands[] = {
6     "AT",
7     "AT+ChannelMode=1",
8     "AT+BAND=8",
9     "AT+DevEui="           ";
10    "AT+AppEui="          ";
11    "AT+AppKey="         ";
12    "AT+JOIN=1,8",
13    //"AT+SEND=1:1:7269767679", // HELLO Message
14    "AT+SEND=1:1:HELLO",
15
16
17 };
18 const int numCommands = sizeof(atCommands) / sizeof(atCommands[0]);
19 int currentCommand = 0;
20
21 unsigned long lastDebounceTime = 0;
22 unsigned long debounceDelay = 50;
23 int lastButtonState = HIGH;
```

## Modifying the LoRa 08H Code...



```
25 void setup() {
26     // Initialize serial communication
27     Serial.begin(9600);
28     Serial1.setRX(1);
29     Serial1.setTX(0);
30     Serial1.begin(9600);
31
32     // Set up LED pin
33     pinMode(LED_PIN, OUTPUT);
34     digitalWrite(LED_PIN, LOW);
35
36     // Set up button pin
37     pinMode(BUTTON_PIN, INPUT_PULLUP);
38 }
39
40 void loop() {
41     // Check button state
42     int reading = digitalRead(BUTTON_PIN);
43
44     if (reading != lastButtonState) {
45         lastDebounceTime = millis();
46     }
```

```
47
48     if ((millis() - lastDebounceTime) > debounceDelay) {
49         if (reading == LOW) { // Button is pressed (active low)
50             sendATCommand();
51         }
52     }
53
54     lastButtonState = reading;
55
56     // Handle incoming serial data
57     if (Serial1.available()) {
58         Serial.write(Serial1.read());
59     }
60 }
```

## Modifying the LoRa 08H Code...



```
61
62 void sendATCommand() {
63     Serial1.println(atCommands[currentCommand]);
64     Serial.print("Sent command: ");
65     Serial.println(atCommands[currentCommand]);
66
67     blinkLED();
68
69     currentCommand = (currentCommand + 1) % numCommands;
70 }
71
72 void blinkLED() {
73     digitalWrite(LED_PIN, HIGH);
74     delay(BLINK_DURATION);
75     digitalWrite(LED_PIN, LOW);
76 }
```

Pressing the Collision Sensor will transmit the "HELLO" message!

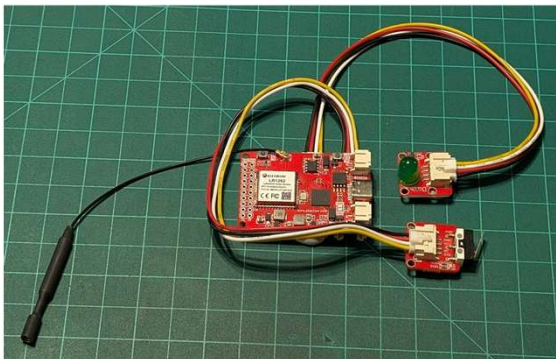
## Question 4

**What devices can be used to check that a message has been received by a Gateway?**

- a) A smartphone or tablet.**
- b) A notepad or smartphone.**
- c) A log or UART serial terminal.**
- d) none of the above**



## Lab: Build and Test a LoRaWAN



```
pi@raspberrypi: ~/LR1302_loraWA_HAL/sx1302_hal/packet_forwarder
File Edit Tabs Help
INFO: Received pkt from mote: 01010101 (fcnt=257)
JSON up: {"rxpk":[{"jver":1,"tmst":3157619354,"chan":1,"rfch":0,"freq":904.100000,"mid":8,"stat":1,"modu":"LORA","datr":"SF10BW125","codr":"4/5","rssi":-27,"lsnr":13.8,"foff":-245,"rssi": -26,"size":23,"data":"AAEBAQEBAQEB5ZwG0H7Vs3BSa8DnGZo="}]}
INFO: [up] PUSH_ACK received in 62 ms
INFO: Received pkt from mote: 01010101 (fcnt=257)
JSON up: {"rxpk":[{"jver":1,"tmst":3158435161,"chan":1,"rfch":0,"freq":904.100000,"mid":8,"stat":1,"modu":"LORA","datr":"SF10BW125","codr":"4/5","rssi":-30,"lsnr":12.5,"foff":-249,"rssi": -29,"size":23,"data":"AAEBAQEBAQEB5ZwG0H7Vs3CpZkd7eNA="}]}
INFO: [up] PUSH_ACK received in 62 ms
INFO: Received pkt from mote: 01010101 (fcnt=257)
```

## Lab: Build and Test the LoRaWAN...



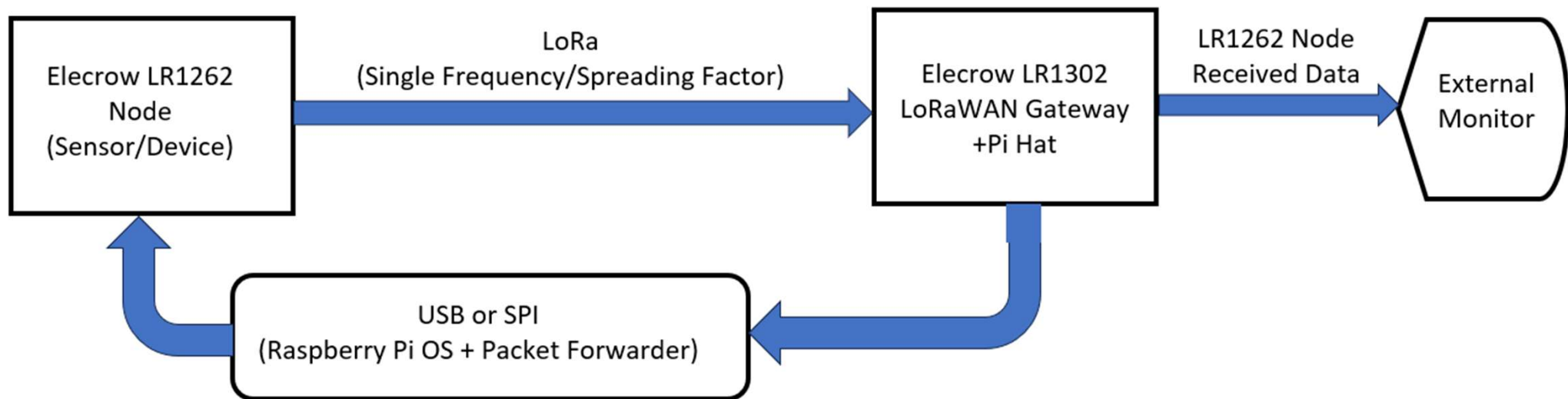
### Participant Learning Objectives:

- Participants will learn to upload the modified LoRa 08H code to the LR1262 Node board.
- Participants will learn to test the LoRaWAN and observe the transmitted operation on an external monitor attached to a Raspberry Pi.
- Participants will learn to add an external switching device to the LR1262 Node board.

## Lab: Build and Test the LoRaWAN...



### Concept System Diagram



Single Channel LoRaWAN Network Architecture Diagram

## Lab: Build and Test the LoRaWAN...

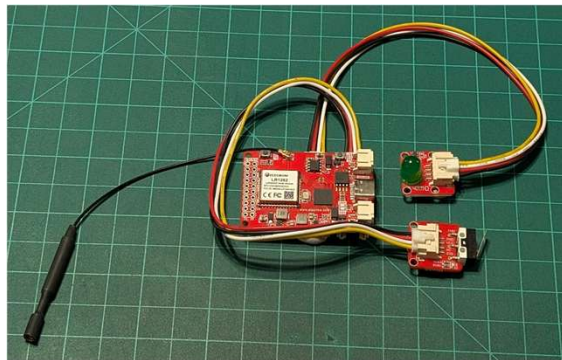


```
Modified_LoRa_08H | Arduino IDE 2.3.6
File Edit Sketch Tools Help
Raspberry Pi Pico
Modified_LoRa_08H.ino
1 #define LED_PIN 3
2 #define BUTTON_PIN 2
3 #define BLINK_DURATION 100 // milliseconds
4
5 const char* atCommands[] = {
6   "AT",
7   "AT+ChannelMode=1",
8   "AT+BAND=8",
9   "AT+DevEui=",
10  "AT+AppEui=",
11  "AT+AppKey=",
12  "AT+JOIN=1,8",
13  //"AT+SEND=1:1:7269767679", // HELLO Message
14  "AT+SEND=1:1:HELLO",
15
16
17 };
18 const int numCommands = sizeof(atCommands) / sizeof(atCommands[0]);
19 int currentCommand = 0;
20
21 unsigned long lastDebounceTime = 0;
22 unsigned long debounceDelay = 50;
```

Ln 1, Col 1 Raspberry Pi Pico on COM6 [not connected]

The Partial  
Modified  
LoRa 08H  
Code

## Lab: Build and Test the LoRaWAN...



Press the  
Collision Sensor  
Lever

### Transmitted LR1262 Node Data

```
pi@raspberrypi: ~/LR1302_loraWA_HAL/sx1302_hal/packet_forwarder
File Edit Tabs Help
INFO: Received pkt from mote: 01010101 (fcnt=257)
JSON up: {"rxpk":[{"jver":1,"tmst":3157619354,"chan":1,"rfch":0,"freq":904.100000,"mid":8,"stat":1,"modu":"LORA","datr":"SF10BW125","codr":"4/5","rssi":-27,"lsnr":13.8,"foff":-245,"rssi":-26,"size":23,"data":"AAEBAQEBAQE5ZwG0H7Vs3BSa8DnGZo="}]}
INFO: [up] PUSH_ACK received in 62 ms
INFO: Received pkt from mote: 01010101 (fcnt=257)
JSON up: {"rxpk":[{"jver":1,"tmst":3158435161,"chan":1,"rfch":0,"freq":904.100000,"mid":8,"stat":1,"modu":"LORA","datr":"SF10BW125","codr":"4/5","rssi":-30,"lsnr":12.5,"foff":-249,"rssi":-29,"size":23,"data":"AAEBAQEBAQE5ZwG0H7Vs3CpZkd7eNA="}]}
INFO: [up] PUSH_ACK received in 62 ms
INFO: Received pkt from mote: 01010101 (fcnt=257)
```

Note: The received data will be encrypted.

## Lab: Build and Test the LoRaWAN...



Watch the  
LoRaWAN in  
Action!

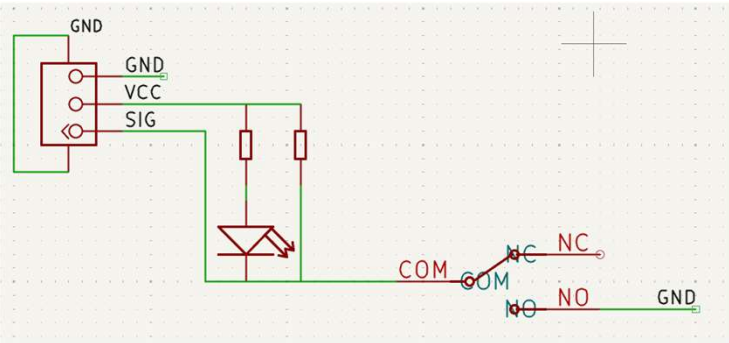
```
andreas@phoenix:~/LRF102/LoRaWAN_HAL/SX1302/LoRaWAN/packet_forwarder$ ./packet_forwarder
File Edit Tabs Help
# PULL_DATA sent: 3 (100.00% acknowledged)
# PULL_RESP(onse) datagrams received: 0 (0 bytes)
# RF packets sent to concentrator: 0 (0 bytes)
# TX errors: 0
### SX1302 Status ###
# SX1302 counter (INST): 3181135381
# SX1302 counter (PPS): 3188797544
# BEACON queued: 0
# BEACON sent so far: 0
# BEACON rejected: 0
### [JIT] ###
src/jitqueue.c:440:jit_print_queue(): INFO: [jit] queue is empty
#-----
src/jitqueue.c:440:jit_print_queue(): INFO: [jit] queue is empty
### [GPS] ###
# GPS sync is disabled
### Concentrator temperature: 27 C ###
##### END #####

JSON up: {"stat":{"time":"2025-06-09 16:38:57 GMT","rxnb":19,"rxok":17,"rxfw":17,
,"ackr":100.0,"dwnb":0,"txnb":0,"temp":26.9}}
INFO: [up] PUSH_ACK received in 62 ms
INFO: [down] PULL_ACK received in 62 ms
```

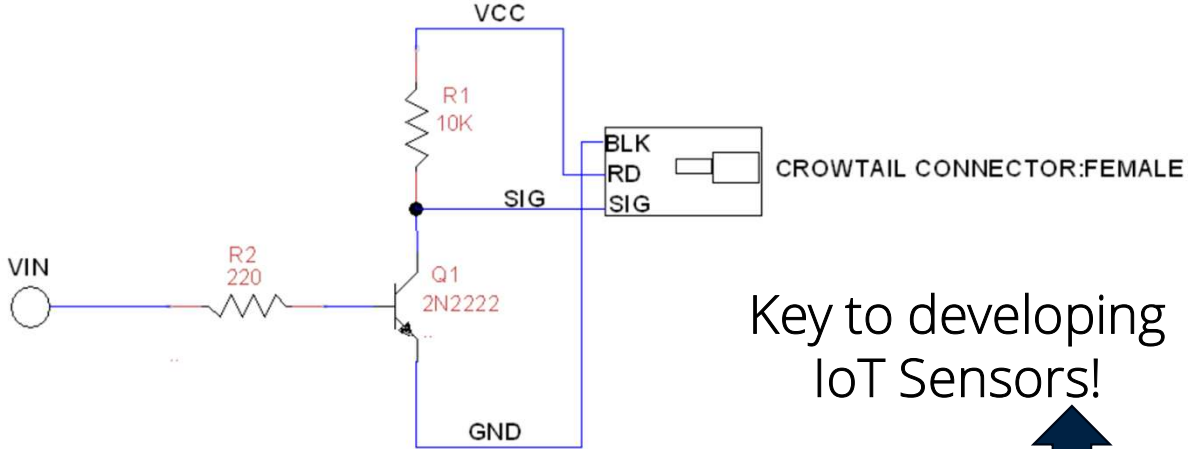


# Lab: Build and Test the LoRaWAN...

Replacing the Collision Sensor with External Control-Sensor Devices



Collision Sensor Electronic Circuit Schematic Diagram:  
Voltage Applied to the "SIG" Input is 0V!



An Electronic Switch Circuit:  
When "VIN" is applied to the circuit, 0V will be applied to the "SIG" Input pin of the Crowtail Connector.

Key to developing IoT Sensors!



## Question 5

**In reviewing slide 38, what voltage is being produced by the circuits shown?**

- a) VCC**
- b) 5V**
- c) 3.3V**
- d) 0V**



## Thank you for attending

Please consider the resources below:

Rahman, H.U., Ahmad, H., Ahmad, M., & Asif Habib, M. (2020). *LoRaWAN: State of the art, challenges, protocols, and research issues*.

[https://www.researchgate.net/publication/348637291\\_LoRaWAN\\_State\\_of\\_the\\_Art\\_Challenges\\_Protocols\\_and\\_Research\\_Issues](https://www.researchgate.net/publication/348637291_LoRaWAN_State_of_the_Art_Challenges_Protocols_and_Research_Issues).

Sani Danladi, M.& Baykara, M. (2022). Design and implementation of temperature and humidity monitoring system using lpwan technology. *Ingenierie des Systemes d'Information*, 27(4), 521-529.

[https://www.researchgate.net/publication/363810711\\_Design\\_and\\_Implementation\\_of\\_Temperature\\_and\\_Humidity\\_Monitoring\\_System\\_Using\\_LPWAN\\_Technology](https://www.researchgate.net/publication/363810711_Design_and_Implementation_of_Temperature_and_Humidity_Monitoring_System_Using_LPWAN_Technology)

## Thank you for attending

Please consider the resources below:

Wilcher, D. (2025). *Hands-on with at commands: The lorawan trainer*. Allaboutcircuits.

<https://www.allaboutcircuits.com/projects/hands-on-with-at-commands-the-lorawan-trainer/>

Wilcher, D. (2025). *Introduction to at commands*. GitHub. [https://github.com/DWilcher/DesignNews-](https://github.com/DWilcher/DesignNews-WebinarCode/blob/main/June_25_Webinar_Code.zip)

[WebinarCode/blob/main/June\\_25\\_Webinar\\_Code.zip](https://github.com/DWilcher/DesignNews-WebinarCode/blob/main/June_25_Webinar_Code.zip)



**DesignNews**

Thank You

Sponsored by

**DigiKey**

