TI Kits

In-Class Activity: TI Kits – Taum Sauk- Part 1

PURDUE UNIVERSITY | School of Engineering Education

In-Class Activity: Overview

Design a simple early-warning system to help prevent a Taum Sauk–style incident. You'll complete TI Kit mini-activities that build toward the final system:

1. Buzzer + Selection Structures
   - use `if/else` structure to create alert patterns

   Class 4B

2. Ultrasonic Ranging
   - measure distance using ultrasonic sensor and read the distance from 4-digit display
   - test system accuracy by examining the % error and SSE
3. Integrated Warning System
   - Combine sensor readings and logic to trigger the buzzer when objects get too close

   Class 5A

4. **Develop an alarm system that could have prevented the Taum Sauk Reservoir accident by detecting dangerously high water levels using the Excel Data Streamer tool.**

   Class 7B

TI Kits

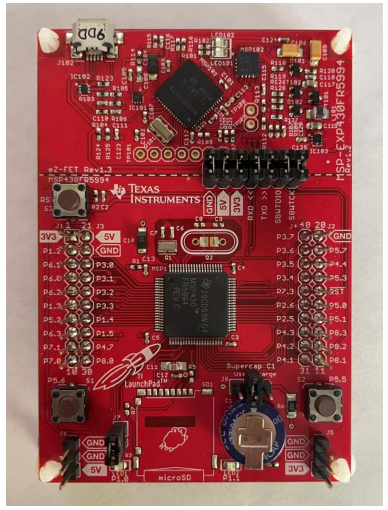
In-Class Activity:

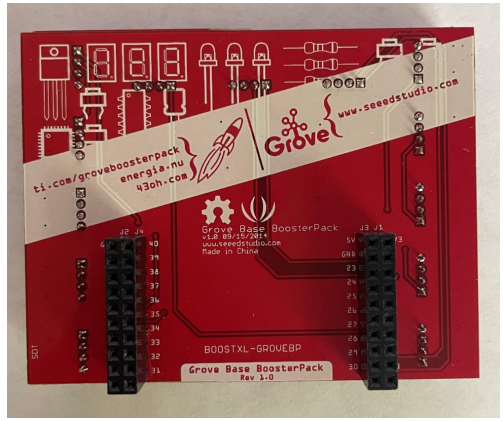
TI Kits – Taum Sauk Part 1

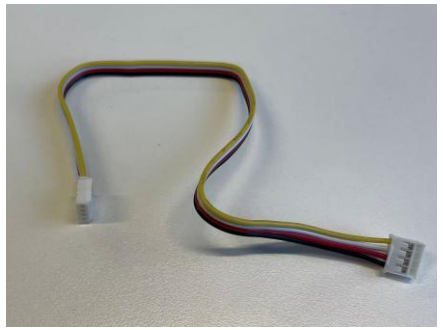
PURDUE UNIVERSITY | School of Engineering Education

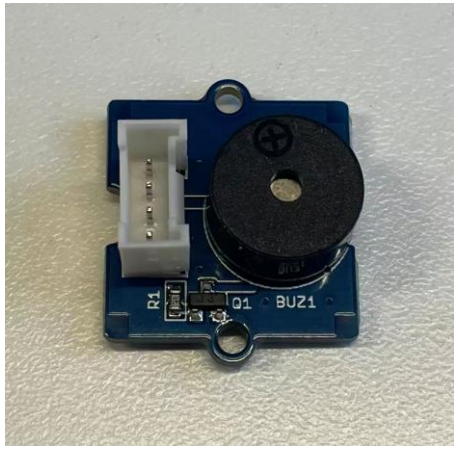
What you will need in this activity:

- TI MSP430 Board
- Grove Base BoosterPack
- 4-Pin Wire
- Buzzer
- Micro B Cable
- Energia
- Download Taum Sauk Part 1 Sketchbook









Introducing Grove board

Grove Base BoosterPack is an add-on board that connects to your TI LaunchPad (MSP430).

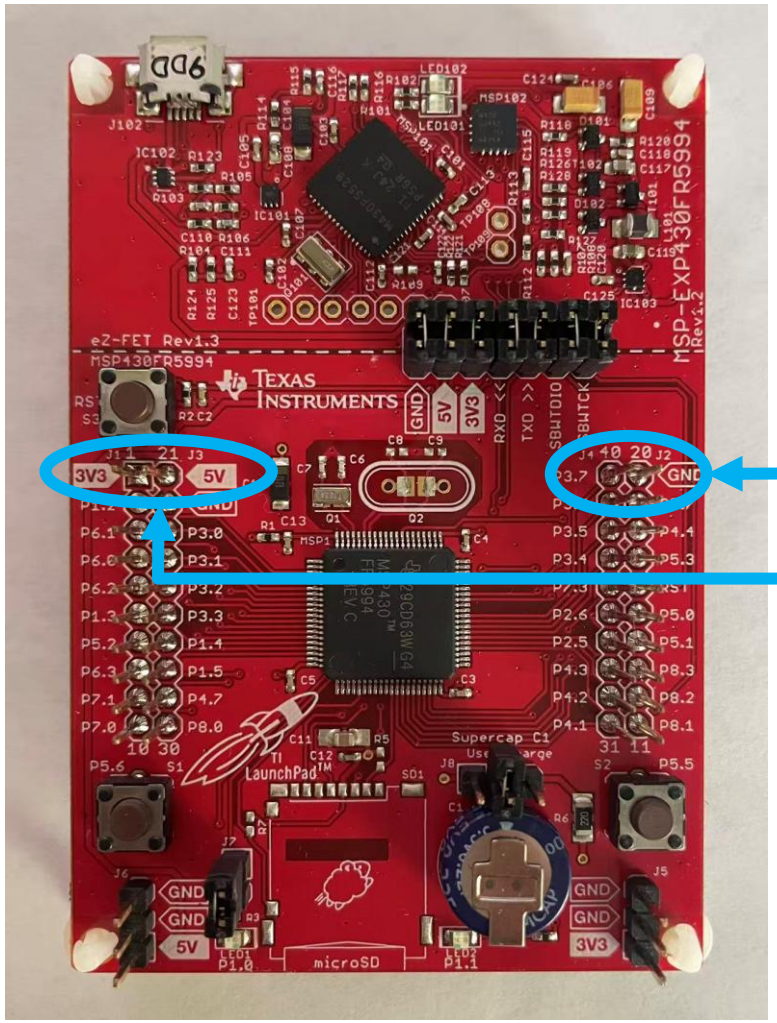
- Think of it as a “sensor playground”: it makes it easy to plug in sensors and actuators without messy wiring.
- Each Grove port uses a 4-pin cable (power, ground, and two signal lines).

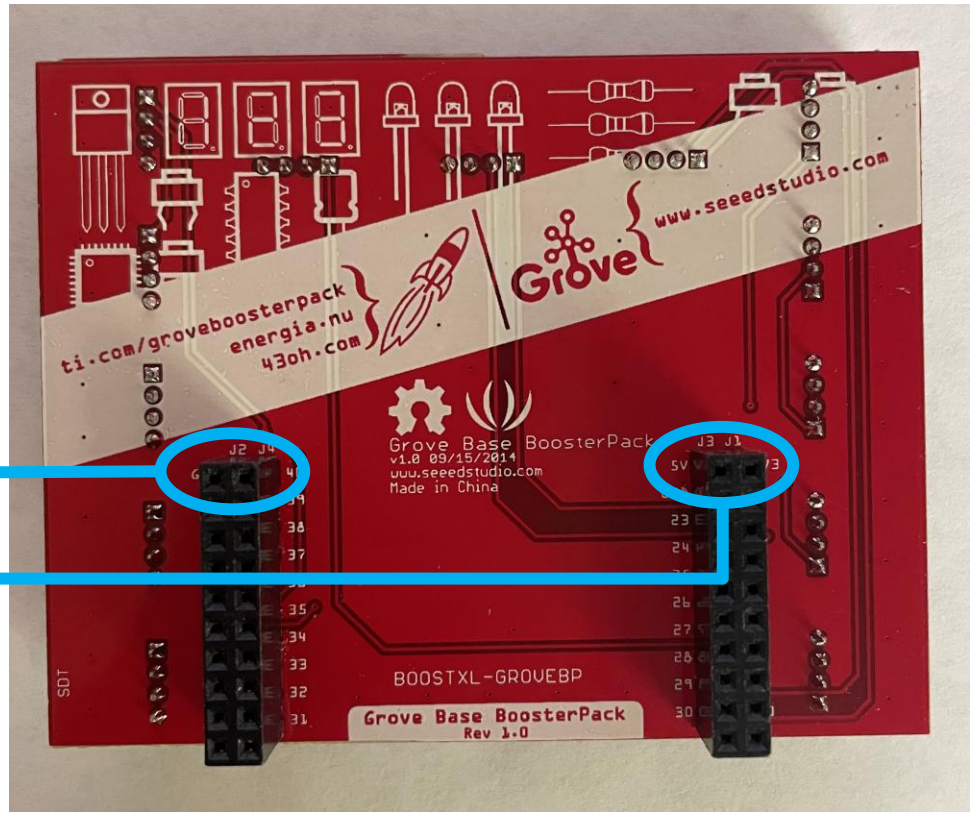
Why do we use it?

- Plug-and-Play: Just connect the 4-pin cable; no breadboard or jumper wire tangle.
- Safe & Consistent: Less chance of wiring mistakes.
- Scalable: You can quickly swap different sensors (buzzer, ultrasonic, light, display, etc.).

How?

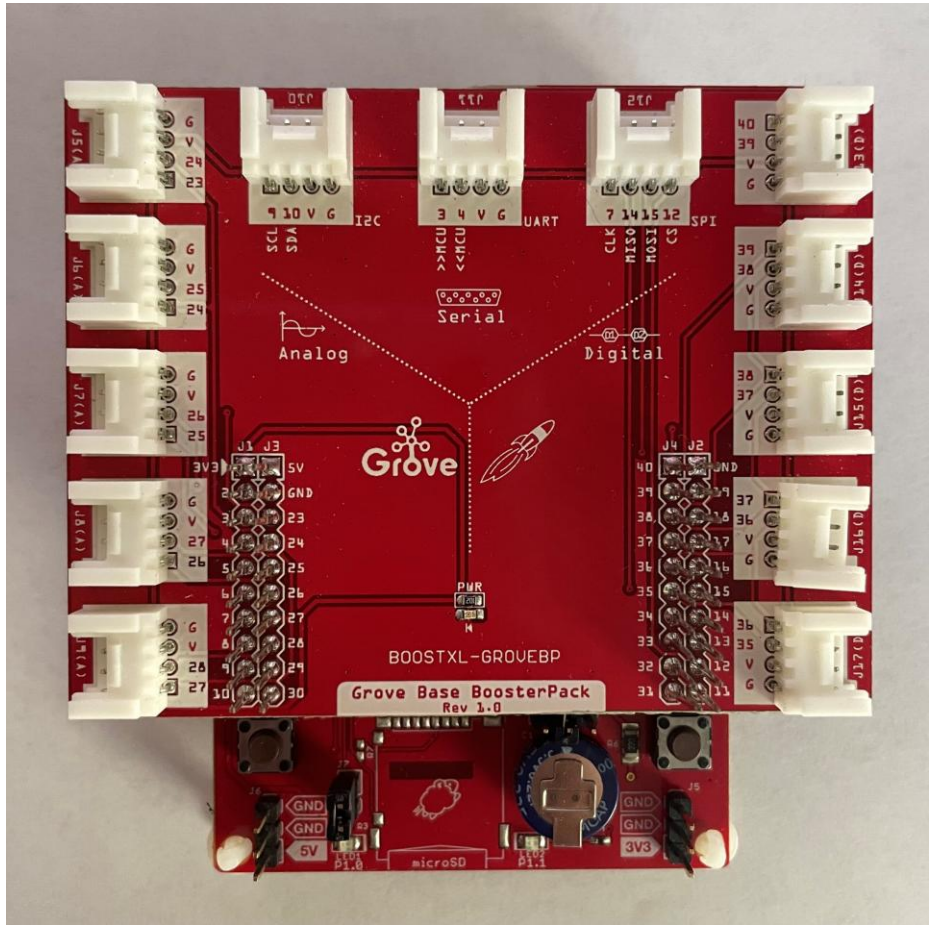
Connecting Grove board to MSP430



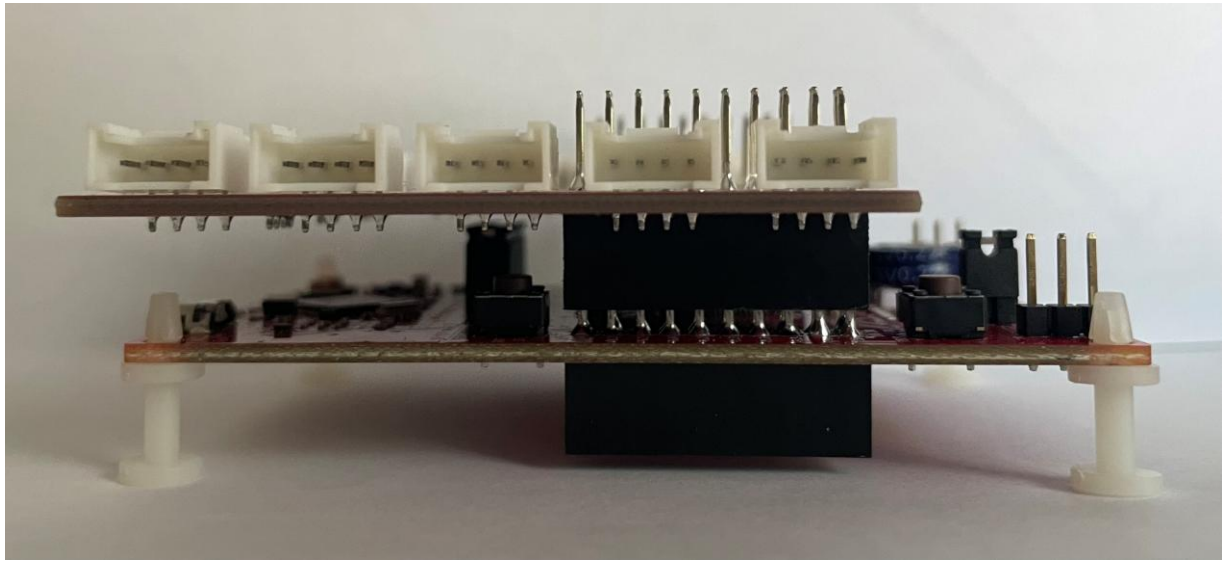


The Grove board snaps on top of the MSP430 LaunchPad, align the pin numbers.

Connecting Grove board to MSP430



Top View



Side View

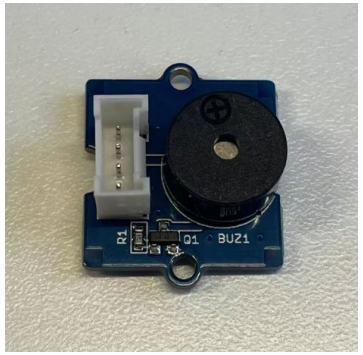
Introducing buzzer

What?

- A buzzer is a simple **electronic device** that **makes sound** when electricity flows through it.
- It can beep, buzz, or play tones depending on how we program it.
- Buzzers are widely used in real-world warning systems (microwaves, alarms, etc.).

Why?

- To give feedback: confirm a button was pressed.
- To warn or alert: when something gets too close or dangerous.
- To add fun: simple tones or patterns.

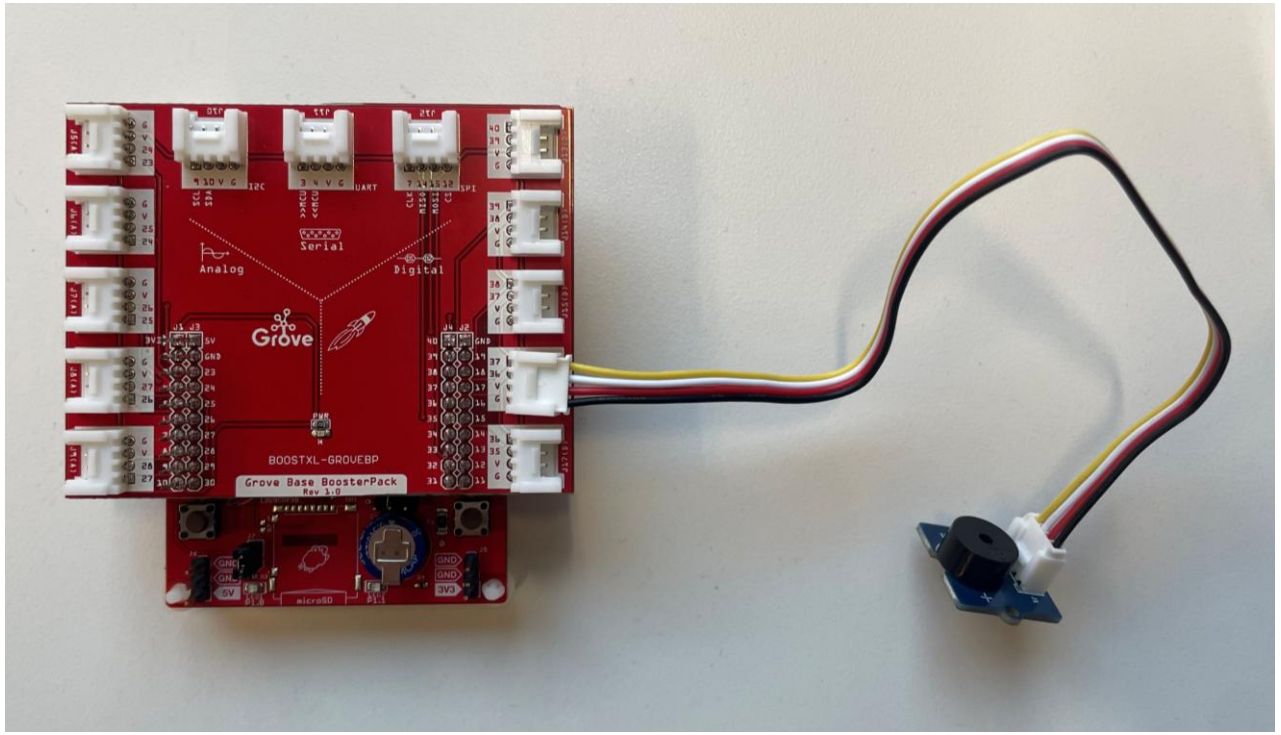


How?

Basic buzzer control

Step 1. Connect the buzzer to the Grove-TI Pack

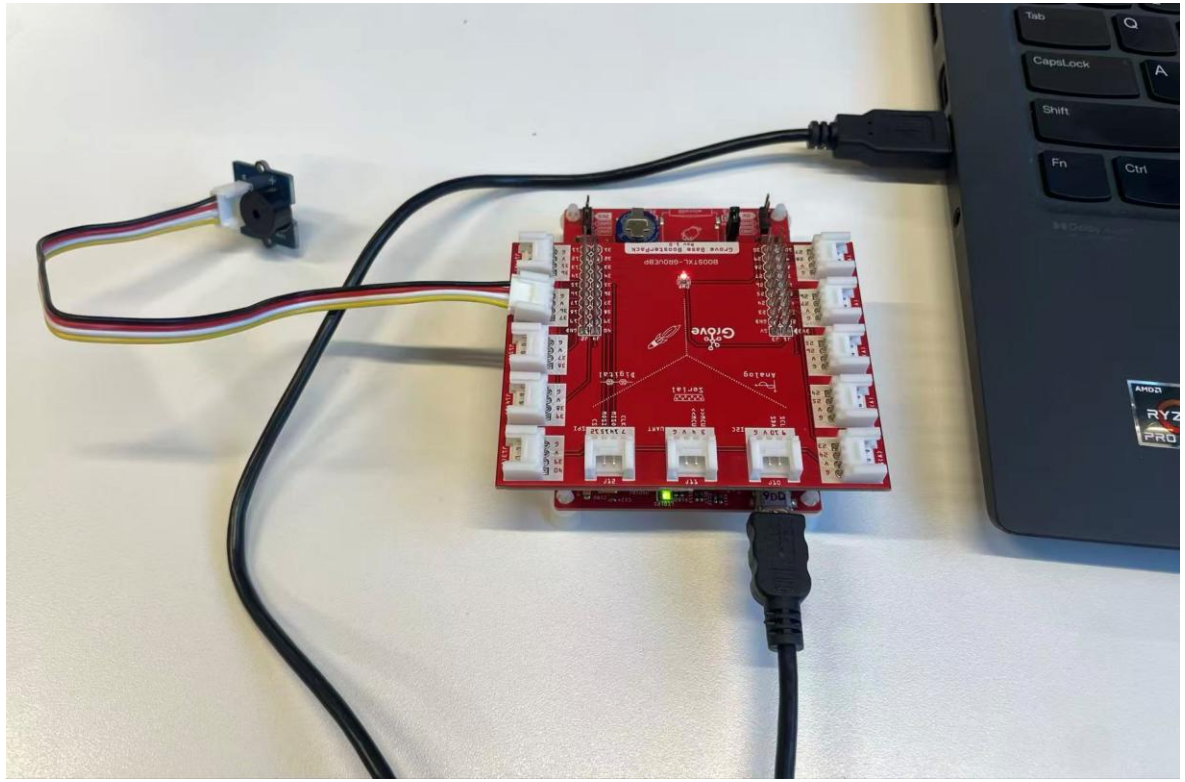
- Note: the buzzer should connect to one of the right side (digital) ports, **record the pin number that the yellow line corresponds to.**



Basic buzzer control

Step 2. Connect the MSP430 to your laptop

- Note: lights on both boards should lit up



Task 1: Draw (by hand or via a computer) a block diagram of the microcontroller set up shown in the picture.

Basic buzzer control

Step 3. Open Energia

- navigate to sketchbook → Taum Sauk Part 1→ buzzer_basic_control.
- modify `buzzerPin` to be the pin number you connected to (read yellow line pin number).

```
int buzzerPin = 37;   // digital pin 37

void setup() {
  pinMode(buzzerPin, OUTPUT);      // set pin as output
  digitalWrite(buzzerPin, LOW);
}

void loop() {
  digitalWrite(buzzerPin, HIGH);   // buzzer ON
  delay(2000);                     // wait 2 seconds
  digitalWrite(buzzerPin, LOW);    // buzzer OFF
  while(true);                     // stop here (no repeat)
}
```

- Verify  , upload  , and see if the buzzer beep for 2 seconds.

… and you could modify the alert pattern…

sample code see next slide

Sample buzzer control

```
int buzzerPin = 27;   // Grove port 27 → digital pin 27

void setup() {
  pinMode(buzzerPin, OUTPUT);
}

void loop() {
  // 1. Turn buzzer ON
  digitalWrite(buzzerPin, HIGH);
  delay(500);   // sound for 0.5 sec

  // 2. Turn buzzer OFF
  digitalWrite(buzzerPin, LOW);
  delay(500);   // silence for 0.5 sec

  // 3. Beep pattern
  for (int i = 0; i < 3; i++) {
    digitalWrite(buzzerPin, HIGH);
    delay(200);   // short beep
    digitalWrite(buzzerPin, LOW);
    delay(200);   // short pause
  }

  delay(1000); // wait before repeating
}
```

The buzzer is plugged into port #27.

The loop function runs forever.

send electricity to the buzzer → it makes sound.

stop sending electricity to the buzzer → stays silent

This is a for loop: it repeats the code inside 3 times.

Sound Result

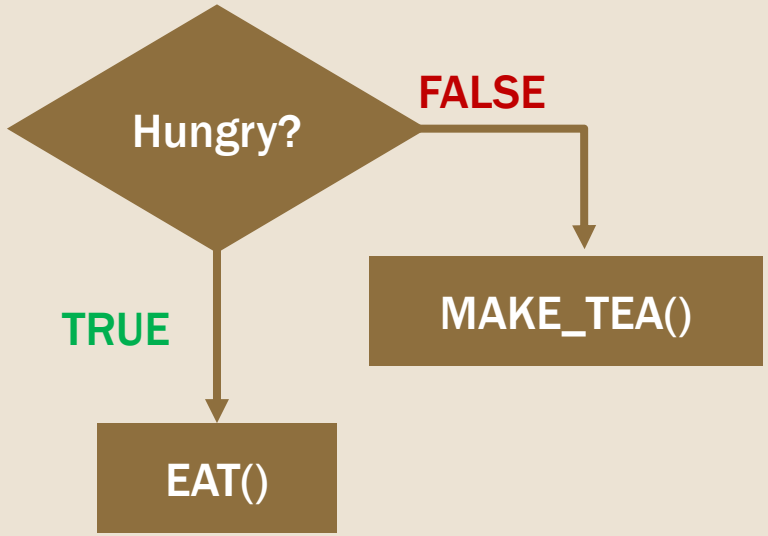
What is a selection structure?

how a program **chooses** between different actions based on a **condition**

Pseudocode:

```
READ hungry
IF hungry == TRUE THEN
        EAT()
ELSE
        MAKE_TEA()
END IF
```

Flowchart:



Energia Syntax:

```
if (hungry == true) {
        EAT();
} else {
        MAKE_TEA();
}
```

Scenario

Imagine you are an engineer working at a hydroelectric dam. The dam holds back a huge reservoir of water. If the water level rises too high, it can overflow or even damage the dam structure, leading to a dangerous accident. Therefore, you want to have a warning system (buzzer) that alerts (beeps) people when the water level reaches a threshold.

- Task 2: Draw a flowchart (by hand or on a computer) that represents the selection structure described in this scenario.

Implementation

In Energia,

- navigate to sketchbook ⟶ Taum Sauk Part 1 ⟶ selection structure
- Task 3: modify the following:
  1. Change value for `buzzerPin` to the port you connected to (if needed).
  2. Add code to the selection structure as indicated.
  3. Change `dist` to above or below `threshold` to test the selection structure.
  4. Copy your working code and paste it to the answer sheet.

Submission Instructions

- Download the two files from here (Activities ->In-Class Activity: Buzzer & Selection Structure)
  - **ENGR131_ICA_Buzzer_YourName.pdf** *(Background and* ***Instructions*** *– do not submit)*
  - **ENGR131_ICA_Buzzer_YourName.docx** *(****Answer Sheet*** *– to complete & submit)*
- Complete the **.docx** file with your answers.
- Submit **only the .docx file** on **Brightspace > In-Class Activity Drop Box.**