

```
In [1]: # Make sure the base overlay is Loaded
from pynq.overlays.base import BaseOverlay
from pynq.lib import Pmod_IO
base = BaseOverlay("base.bit")
```

```
In [2]: from pynq.lib.arduino import Arduino_Analog
from pynq.lib.arduino import ARDUINO_GROVE_A1
from pynq.lib.arduino import ARDUINO_GROVE_A2
from pynq.lib.arduino import ARDUINO_GROVE_A3
pmod_pin2 = Pmod_IO(base.PMODA, 2, 'out')
pmod_pin3 = Pmod_IO(base.PMODA, 3, 'out')
analog1 = Arduino_Analog(base.ARDUIAO,ARDUINO_GROVE_A1)
analog2 = Arduino_Analog(base.ARDUIAO,ARDUINO_GROVE_A2)
analog3 = Arduino_Analog(base.ARDUIAO,ARDUINO_GROVE_A3)
```

```
In [3]: import requests
import random
import time

# Define the API URL for RTD sensor and joystick
url = "https://api.thingspeak.com/update"

# Define the API key
api_key = "ZWDDD7SRLM2MQYX3"
# Define the API URL for reading data
read_url = "https://api.thingspeak.com/channels/2453222/feeds/last.json?pi_key=ZWDDD7SRLM2MQYX3"
```

```
In [ ]:
```

```
In [ ]: import requests
import time

# Define the API URLs for reading data and sending results
read_url = "https://api.thingspeak.com/channels/2453222/feeds.json?results=" # Fetch only the Latest record
write_url = "https://api.thingspeak.com/update" # Replace with your actual API endpoint

# Define the API key
api_key = "ZWDDD7SRLM2MQYX3"

def process_data():
    try:
        # Make the API request to read data
        response = requests.get(read_url)

        # Check if the request was successful
        if response.status_code == 200:
            # Extract the JSON data from the response
            data = response.json()

            # Extract the latest feed from the response
            feeds = data["feeds"]

            # Initialize variables to store the latest values of field3 and field4
            latest_field3 = None
            latest_field4 = None

            # Iterate through the feeds in reverse order to find the latest values of field3 and field4
            for feed in reversed(feeds):
                if "field3" in feed and feed["field3"] is not None:
                    latest_field3 = feed["field3"]
                    break

            for feed in reversed(feeds):
                if "field4" in feed and feed["field4"] is not None:
                    latest_field4 = feed["field4"]
                    break

            # Check if both field3 and field4 have valid values
            if latest_field3 is not None and latest_field4 is not None:
                print(f"Latest values - field3: {latest_field3}, field4: {latest_field4}")

                # Check the conditions and print messages accordingly
                if latest_field3 == "1" and latest_field4 == "1":
                    print("Field3 is 1 and Field4 is 1")
                    pmod_pin2.write(1)
                    pmod_pin3.write(1)
                elif latest_field3 == "1" and latest_field4 == "0":
                    print("Field3 is 1 and Field4 is 0")
                    pmod_pin2.write(1)
                    pmod_pin3.write(0)
                else:
                    print("Field3 is 0")
                    pmod_pin2.write(0)
                    pmod_pin3.write(0)

                # Define the payloads for each field
                payload1 = {"api_key": api_key, "field5": latest_field3}
                payload2 = {"api_key": api_key, "field6": latest_field4}

                # Send data for field5
                response1 = requests.post(write_url, data=payload1)
                time.sleep(20)
                if response1.status_code == 200:
                    print(f"Value {latest_field3} was successfully written to field5.")
                else:
                    print(f"Error writing value to field5. Status code: {response1.status_code}")

                # Send data for field6
                response2 = requests.post(write_url, data=payload2)
                time.sleep(20)
                if response2.status_code == 200:
                    print(f"Value {latest_field4} was successfully written to field6.")
                else:
                    print(f"Error writing value to field6. Status code: {response2.status_code}")

            else:
                print("No valid values found for field3 and field4 in the latest feeds.")

        else:
            print(f"Failed to fetch data from ThingSpeak. Status code: {response.status_code}")

    except Exception as e:
        print(f"An error occurred: {e}")

# Main Loop
while True:
    process_data()
    time.sleep(60) # Adjust the time interval as needed
```

Latest values - field3: 1, field4: 1
Field3 is 1 and Field4 is 1
Value 1 was successfully written to field5.
Value 1 was successfully written to field6.
Latest values - field3: 1, field4: 1

```
Field3 is 1 and Field4 is 1
Value 1 was successfully written to field5.
Value 1 was successfully written to field6.
Latest values - field3: 1, field4: 1
Field3 is 1 and Field4 is 1
Value 1 was successfully written to field5.
Value 1 was successfully written to field6.
Latest values - field3: 1, field4: 1
Field3 is 1 and Field4 is 1
Value 1 was successfully written to field5.
Value 1 was successfully written to field6.
Latest values - field3: 1, field4: 1
Field3 is 1 and Field4 is 1
Value 1 was successfully written to field5.
Value 1 was successfully written to field6.
```

In []:

In []: