SMART QUBE

Internet of Things

Atharva Jakkanwar
jakkanwar.@husky.neu.edu

Smart Qube

A non-intrusive trigger hub for IoT connected devices.

People are usually skeptical of connected devices that are “always listening” or “always looking” using intrusive sensors like microphones or cameras. The goal of this project was to make a prototype of a non-intrusive sensor/trigger hub that can be used to trigger various tasks over IFTTT API calls.

Since the device itself does not connect to the internet, it is isolated from the network when not in range of the central node which helps to add a physical layer of security.

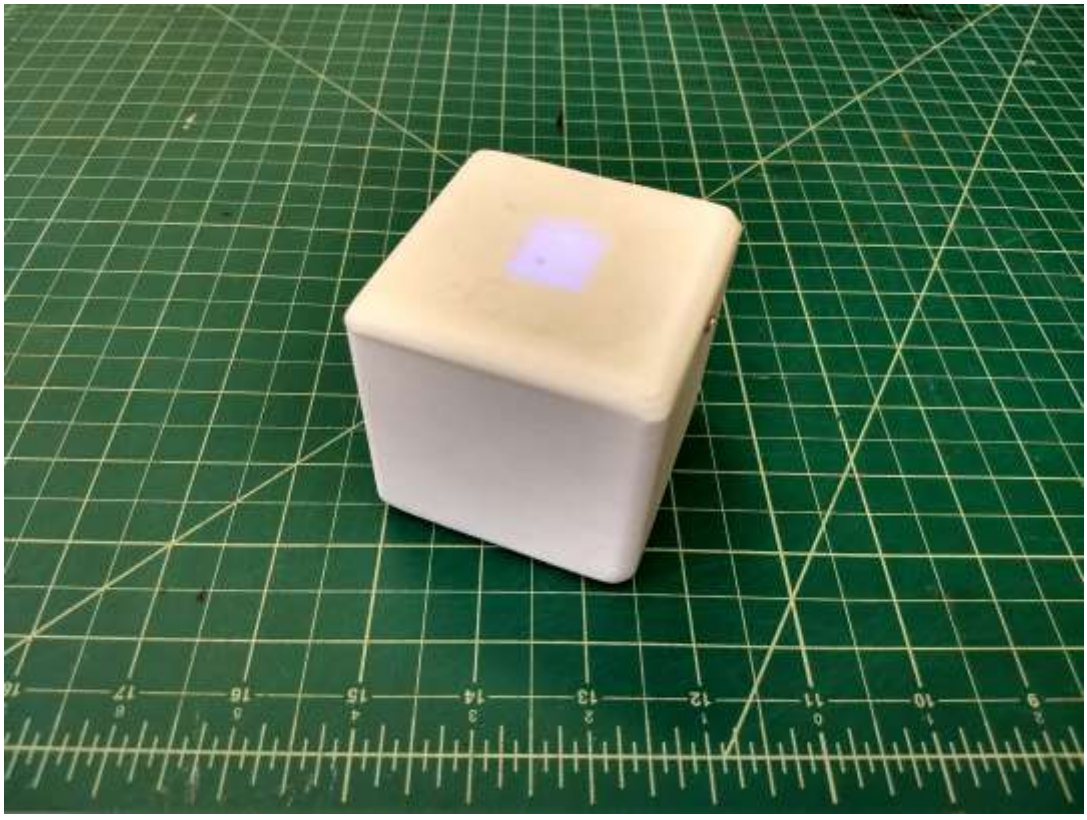
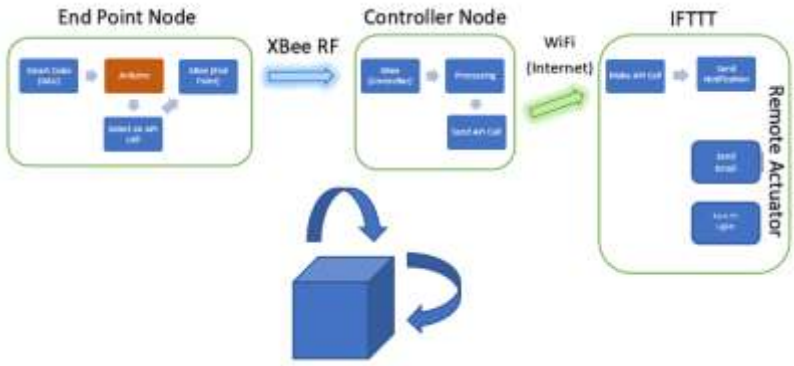


Figure 1. Smart Qube

System Diagram



Implementation details

The device comprises of the case along with the electronics contained within the case.

At the heart of the device is an *Arduino Nano V3* that orchestrates all the other sensors and actuators. The IMU (Inertial Measurement Unit) *MPU 6050* is the sensor that measures the rotation and orientation of the device. The cube wirelessly communicates to a central node (A computer in this case) using the *XBee S1* module. The way the communication protocol is set up in such a manner that the device will transmit data only if the central hub is present to receive the data. After each packet of data is received, the central node sends back a special character that prompts the device to send the next packet. If the central node is missing, the device won't send any data packets.

To avoid having to set up all the different actions that the cube can react to, on the Arduino itself, the cube sends out raw Yaw, Pitch and Roll values to the central node where they can be processed in whichever way necessary. Based on this processing (like identifying change of cubes rotation from clockwise to anticlockwise) the program on the central node then sends out an API call to any one of the connected devices associated with that motion. This also makes it possible to add new devices the cube without having to reprogram the Arduino.

To prevent accidental triggers, the device will start transmitting only if the touchpad on the top of the cube is activated.

Parts List

	DEVICE	PURPOSE
MICROCONTROLLER	1. ARDUINO NANO V3	MICROCONTROLLER
INPUT SENSORS	1. ACCELEROMETER / GYRO MPU 6050	ORIENTATION SENSOR
	2. CAPACITIVE TOUCH SENSOR HELTEC TOUCH PAD	PREVENT ACCIDENTAL ACTUATION
OUTPUT ON DEVICE	1. PIEZO BUZZER	AUDIO NOTIFICATION OF A TRIGGER
	2. HELTEC TOUCH PAD	TOUCH NOTIFICATION
COMMUNICATION	1. XBEE S1	COMMS TO/FROM DEVICE
	2. XBEE S1 WITH EXPLORER SHILED	RECEIVER FOR THE CENTRAL NODE

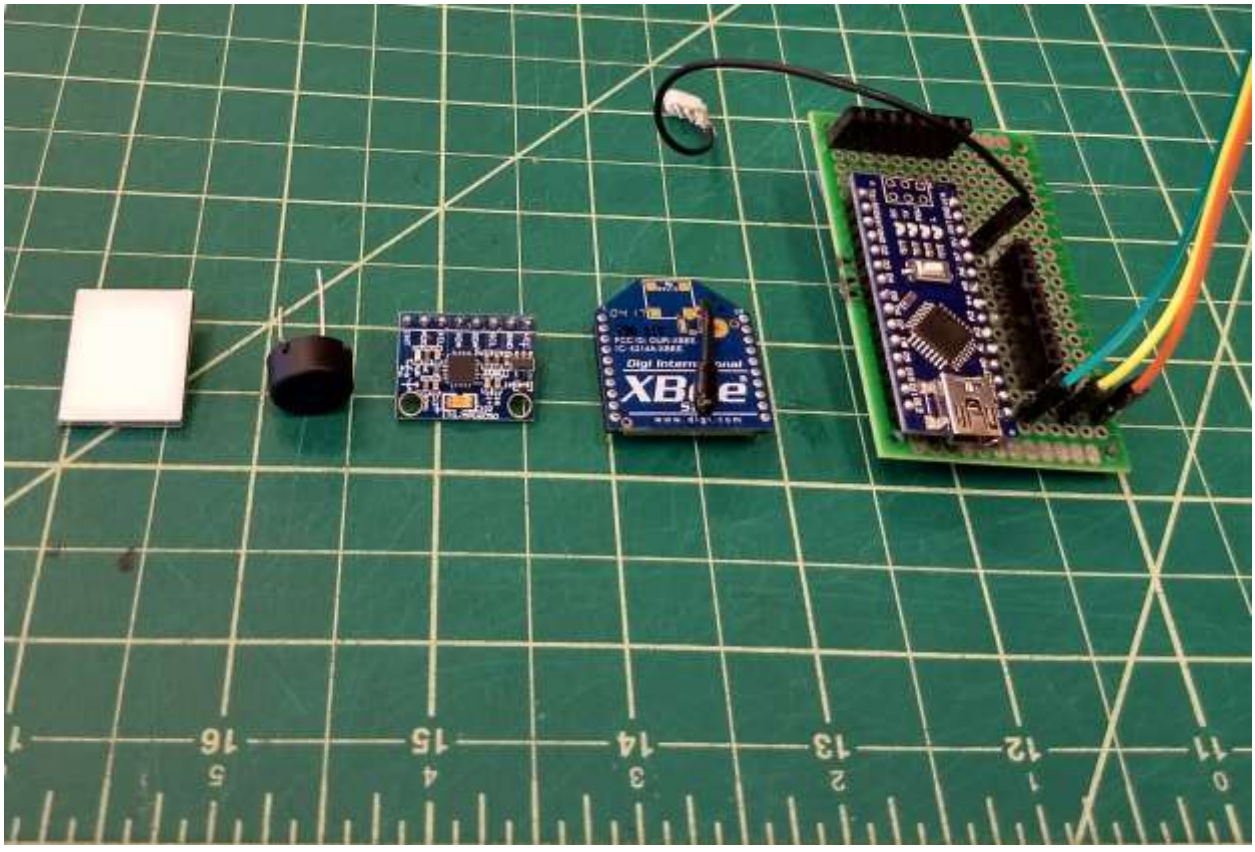
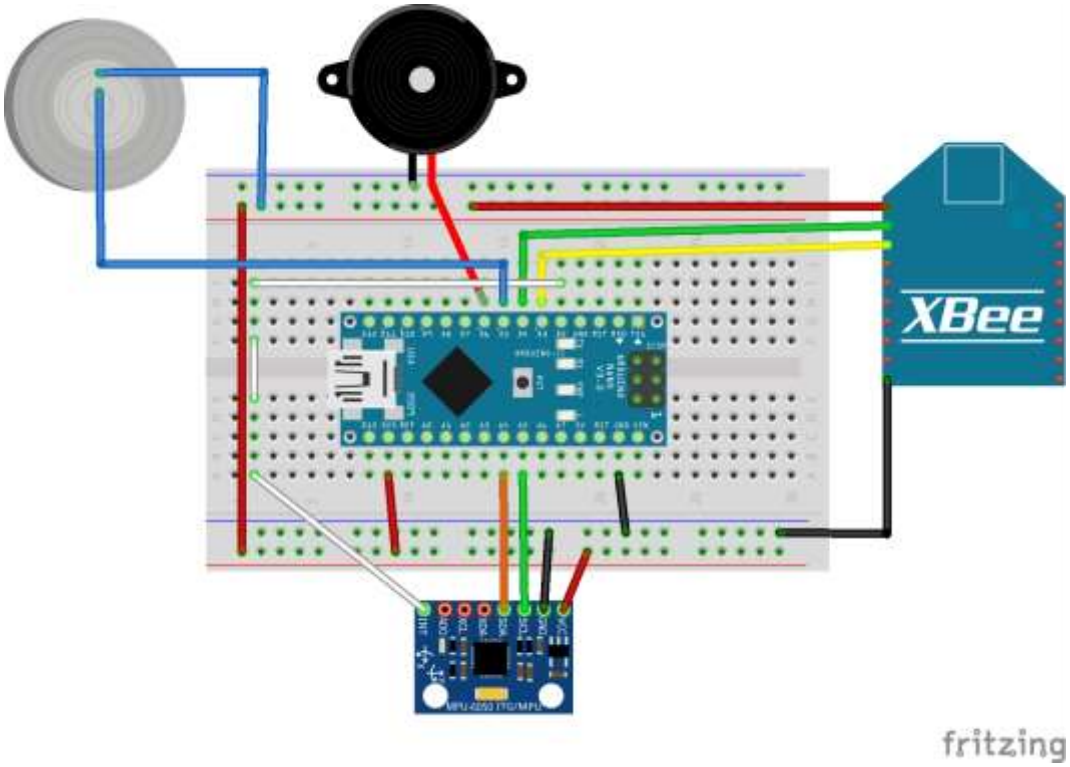


Figure 2. Touch Pad, Piezo Buzzer, MPU6050, XBee S1, Arduino Nano on a protoboard with connectors

Circuit diagram



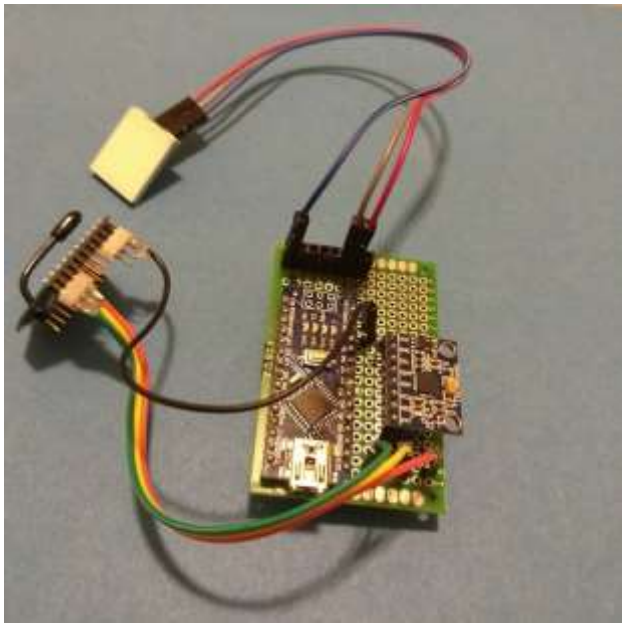


Figure 3. The assembled circuit

Code

The barebones code to run MPU605 with Arduino can be found in the libraries provided at: https://github.com/jrowberg/i2cdevlib

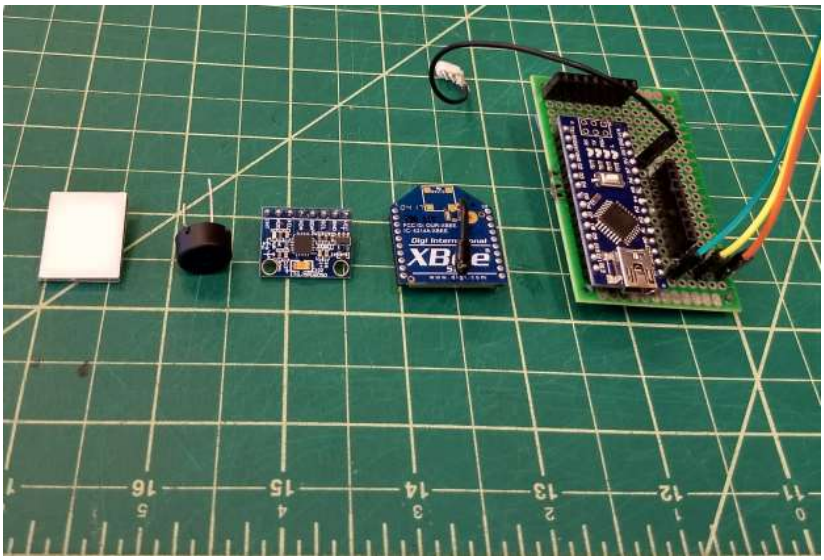
The code for the actual Arduino implementation can be found at: https://github.com/atharvajakkanwar/smart-qube

The code for the central node is written in Processing language and can be found at: https://github.com/atharvajakkanwar/smart-qube

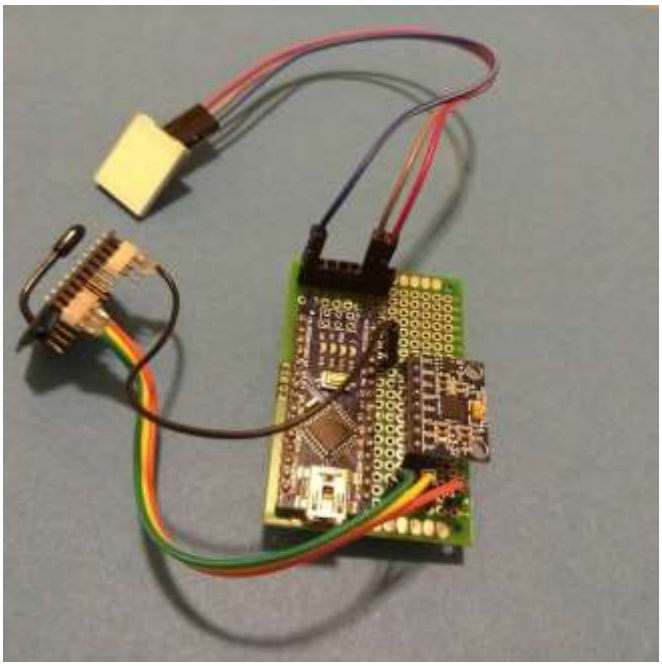
Development Process

The idea was to create a connected device that people would want to have without worrying about the privacy issues associated with it. I wanted it to be minimal yet functional, with as little as possible actuation involved from a human. The most intuitive way other than speaking or video recording (both of which have privacy concerns) was gestures. Since cameras were not to be used to record gesture, an accelerometer was the next best choice.

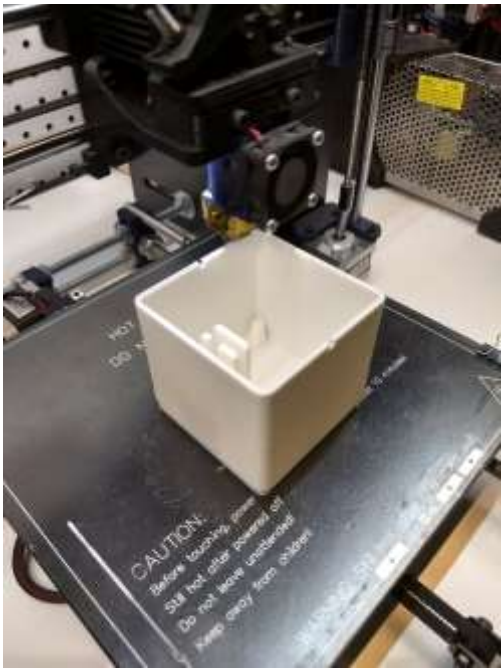
Obtaining the parts:



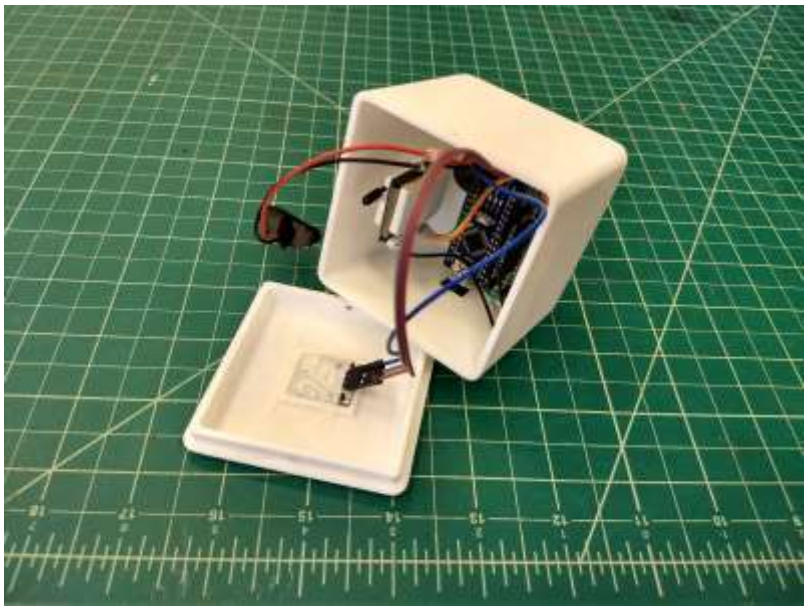
Deciding on the circuit, making it as compact as possible. I shifted from Arduino UNO to Arduino Nano:

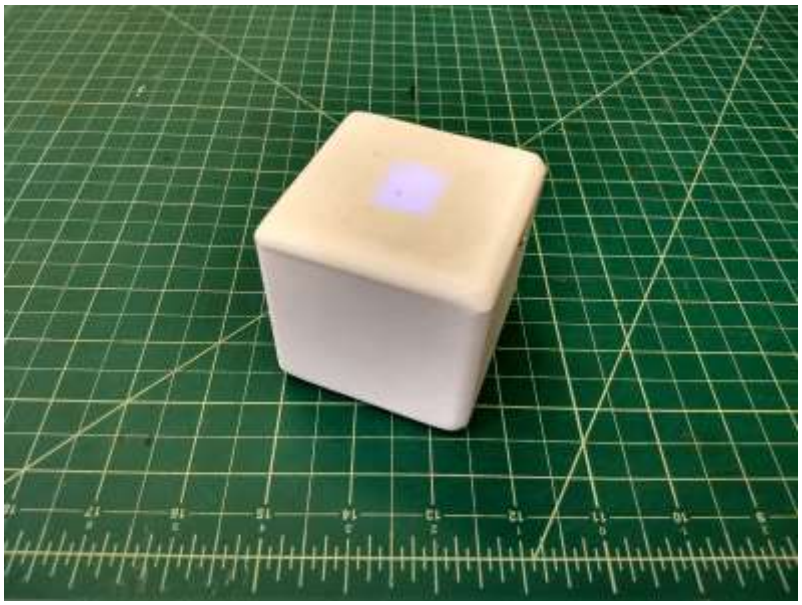


Designing and 3d printing the case:



Assembling it all together with no components visible outside:





Next Steps

- Analog triggers:
  As of now, the code only deals with API calls based on changes in orientation. A simple modification to the code can make the cube respond to the rotation motion as well, mapping the angular change to a function like change in volume of a speaker or increasing brightness of a light.

- A device on each face:
  As a prototype, only one of the faces of the cube (one direction) is associated with a device. We can assign up to 4 functions to this particular face. Minor changes in code and a few more touch sensors can lead to 6 devices associated with 6 faces of the cube (the number is limited only by the number of faces that the case can have).

- A better PCB design:
  A better circuit fabrication could lead to an even more compact and sturdy design.

- Power source:
  The power source used for the prototype is a standard 9v battery. A LiPo battery could not be used as the standard LiPo batteries in small form factor are 3.7v – 5.4v whereas the Arduino required over 7V to function without overclocking. More study into the power source and battery can possible lead to a rechargeable device.

- IFTTT calls:
  IFTTT API calls take a few second to go through. This causes a delay of about 2-3 seconds after the action is taken. More IoT communication protocols can be considered.

Working Demo Video: YouTube