École Polytechnique fédérale de Lausanne

MT-MA2 - Spring 2020

Documentation Kitchen Sensors

Semester Project

Ramez Iskandar

EPFL

Mathis Group
computational neuroscience & AI

Table of contents

Introduction

The sensorization of the smart kitchen was separated into 2 main parts : Wireless and Wired IMU sensors. The Wireless sensors were used on all mobile kitchen objects, e.g. Utensil, bottles,etc. The wired one were put on fixed furniture, e.g. Cabinet doors and drawers. Here will present the information needed to reconstruct a similar system, how to run it and how to process the data.

Hardware

Wireless Sensors

Wireless sensors are used on the mobile objects in the kitchen, e.g. Bottles, knives, etc. The sensors used are the AX3[1] sensors from Axivity (*Datasheet*[2]). These IMU sensors record the acceleration along the X, Y and Z axis.They are waterproof and can last around 2 weeks with one charge.

Wired Sensors

Wired sensors are used on fixed furniture in the kitchen : Cabinet doors, drawers and fridge door. The main components of the system are an Arduino Uno and the 9 DOF sensors.

We see on Figure 1 a detailed description of the system/wiring.

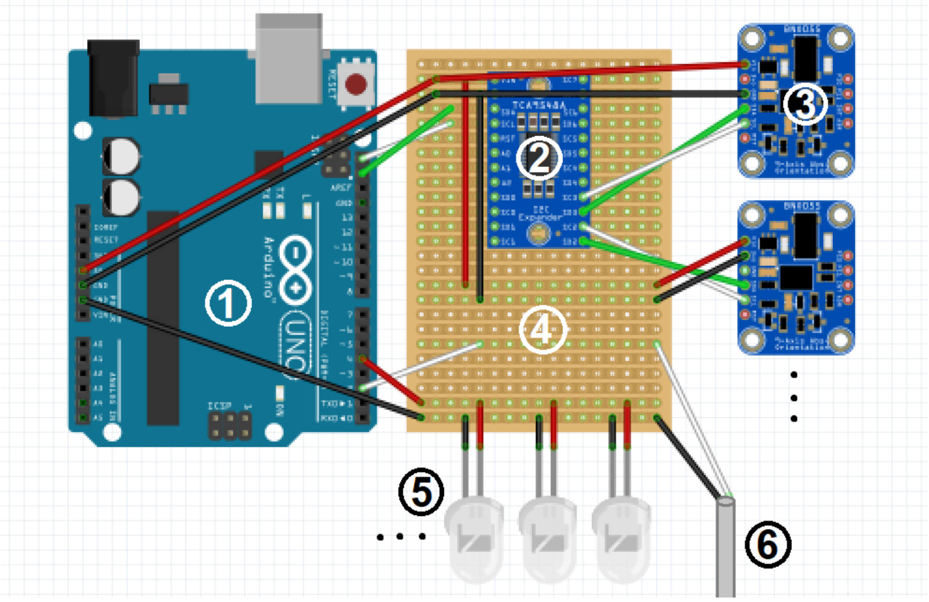

FIGURE 1 – Caption

[1] https://axivity.com/product/ax3

[2] https://axivity.com/files/resources/AX3_Data_Sheet.pdf

A list of all the used components can be found below :

1. 1x Arduino Uno [3]
2. 1x Adafruit TCA9548A 1-to-8 I2C Multiplexer [4]
3. 8x Adafruit BNO055 Absolute Orientation Sensor [5]
4. Stripboard
5. 6x White LED
6. Audio cable from Azure Kinect [6]

The BNO055 sensors use the I2c buses to communicate, and since only one I2C bus is available on the Arduino, we need a multiplexer : Adafruit TCA9548A 1-to-8 I2C Multiplexer. This multiplexer allows to have 8 different sensors on this one I2C bus. To wire the sensors, we need to connect the SDA and SCL of the sensors with the corresponding ones in the multiplexer. Then the SDA and SCL pins in the multiplexer needs to be wired to the corresponding one in the Arduino. The Vin and GND pins of the sensors and multiplexer needs to be respectively wired to the 5V and GND of the Arduino.

For the audio cable, 3 cables were extracted from this one : Red and White cables (representing Right and Left) and a Black cable (representing the ground). The White cable need to be connected to Pin 2 on the Arduino and the black cable connected to GND.

Each LED has to legs, the shorter one represents the ground. The black cable (ground) needs to be wired to the GND pin, and the other red cable to Pin 4.

Recording

Wireless Sensors

Before starting an experiment, we need to launch each sensor by connecting it to the computer and start the recording. The sensor can then record data for up to 2 weeks. To retrieve the recordings, we'll then need to connect the sensor to the computer and download the data. This is easily done thanks to the AX3 GUI.

A detailed description on the usage of the AX3 sensors (recording, downloading data, etc) can be found here : `https://github.com/digitalinteraction/openmovement/wiki/AX3-GUI`. Some videos of how to use the GUI can also be found here : `https://www.youtube.com/user/openmovementncl`

We record the data at 100Hz and download the recordings using the **Export Raw CSV** command and choose **Formated(Y-M-D h :m :s.f** under Timestamp Estimation.

[3] `https://docs.arduino.cc/resources/datasheets/A000066-datasheet.pdf`

[4] `https://cdn-learn.adafruit.com/downloads/pdf/adafruit-tca9548a-1-to-8-i2c-multiplexer-breakout.pdf`

[5] `https://cdn-learn.adafruit.com/downloads/pdf/adafruit-bno055-absolute-orientation-sensor.pdf`

[6] `https://docs.microsoft.com/en-us/azure/kinect-dk/multi-camera-sync`

Wired Sensors

For the wired sensors, we need 2 software to be able to record the desired data : Arduino IDE[7] and Processing IDE[8].

A detailed explanation of how to run and save the data can be found in this video : `https://github.com/amathislab/SmartKitchen_Sensors/blob/main/Instructions_video.mp4`

Synchronization

Wireless Sensors

To synchronize the wireless sensors with the wired system, we use the date. Since the wireless sensors work at higher frequencies, we can extract for each date value in the wired system it's corresponding one in the wireless system. Thus, we get a new dataset for the wireless sensors with only readings corresponding to the ones in the wired system. This is done using the code *sync_save.py*

Wired Sensors

The wired sensors are synchronized with the Azure Kinect camera using the audio cable. From the audio cable coming out of the camera, we are able to extract the clock of the cameras (train of pulses representing one frame at each pulse). By plugging this pulse train in the sensors system, we are able to tell it to read only a value when there is a pulse, this way we have sensors recordings that match the camera frame.

With these two steps, we get a fully synchronized system.

[7] `https://www.arduino.cc/en/software/OldSoftwareReleases`

[8] `https://processing.org/download`