# Additional Sensor Setup

This document describes how to incorporate an additional sensor that gives analog or digital output. The method described below modifies the library files coming with the Arduino turtlebot3 package.

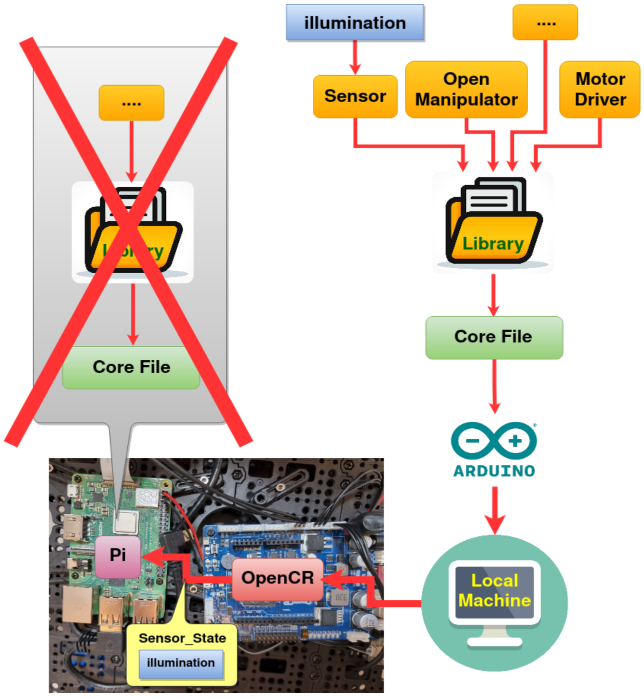
The general logic is that the OpenCR board can only take one Arduino file and in our case, such file is

- Turtlebot waffle without arm: `Library/Arduino15/packages/OpenCR/hardware/OpenCR/1.4.19/libraries/turtlebot3/examples/turtlebot3_waffle/turtlebot3_core.ino`
- Turtlebot waffle with arm: `Library/Arduino15/packages/OpenCR/hardware/OpenCR/1.4.19/libraries/turtlebot3/examples/turtlebot3_with_open_manipulator/turtlebot3_with_open_manipulator_core.ino`

The core file will call a bunch of other files from the library to incorporate functionalities like open arm manipulator, sensor state, etc. Here, we notice that the illumination sensor in the sensor_state library is very similar to the sensor information we want to get so we decide to modify that library file to incorporate the additional sensor:

- File path: `Library/Arduino15/packages/OpenCR/hardware/OpenCR/1.4.19/libraries/turtlebot3/src/turtlebot3/turtlebot3_sensor.cpp`
- Function name: `Turtlebot3Sensor::getIlluminationData()`

## **Diagram**:

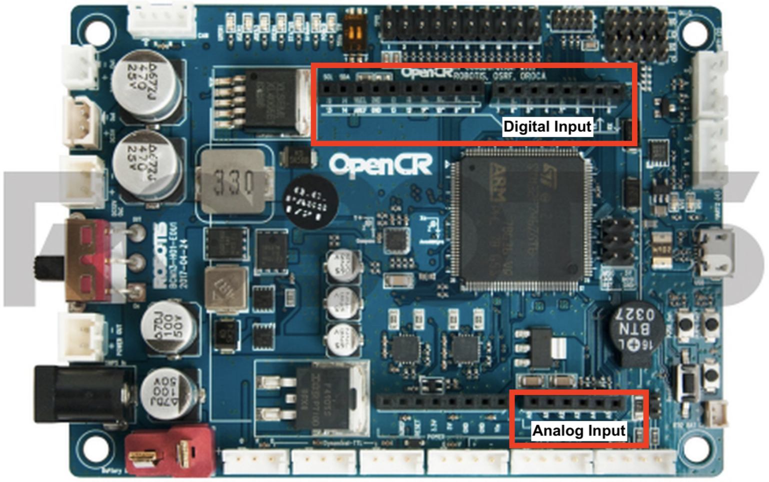


## Setup steps:

1. Install the Arduino IDE on your local machine (not VM) follow the link below:
    a. https://emanual.robotis.com/docs/en/parts/controller/opencr10/#install-on-linux
    b. ***Step 4.1.3 – 4.1.5.2*** for Linux machines
    c. ***Step 4.2.1 – 4.2.3.2*** for Mac machines
    d. Nobody cares about Windows machines ;)

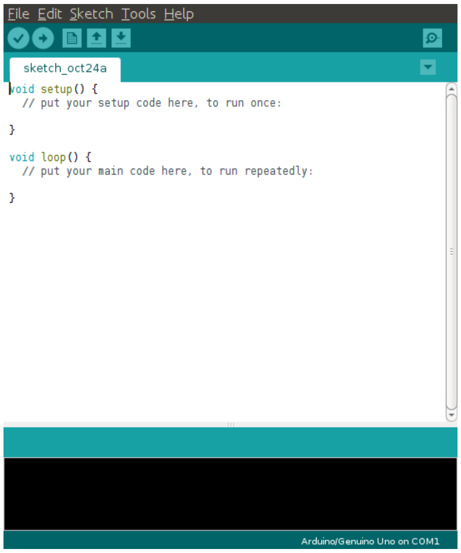
2. USB Port setting
    a. This is ***Step 4.1.5.3*** for Linux or ***Step 4.2.3.3*** for Mac.
    b. Disconnect OpenCR from Pi and connect OpenCR to your computer
    c. If no port detected, try Recovery mode (***Step 8.1***) follow the link below: https://emanual.robotis.com/docs/en/parts/controller/opencr10/#recovery-mode

3. Connect sensor to OpenCR board
    a. For example, for pressure sensor code, you can find on keyestudio web: https://wiki.keyestudio.com/Ks0309_Keyestudio_Thin-film_Pressure_Sensor_(Black_and_Eco-friendly)
    b. For the S pin, if the sensor you want to connect gives digital/ binary data, connect it to Digital Input; if it gives analog data, connect it to Analog Input.
    c. The sensor also needs to be connected to + (5V) and - (ground) pins.
    d. This is how OpenCR board looks like:



4. Write Arduino code for your sensor and test if it's working:

a. The Arduino IDE interface where you will write and test your code:



```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

b. For example, for pressure sensor code, you can find on keyestudio web: [https://wiki.keyestudio.com/Ks0309_Keyestudio_Thin-film_Pressure_Sensor_(Black_and_Eco-friendly)](https://wiki.keyestudio.com/Ks0309_Keyestudio_Thin-film_Pressure_Sensor_(Black_and_Eco-friendly)) (There is actually an error in the code they provided, see below)

c. For receiving binary outputs, we connect the **S pin** to Digital In (with numbers) on OpenCR board, and write `pinMode(pin_number,INPUT)` in function `setup()` and `digitalRead(pin_number)` in function `loop()`

d. For receiving analog outputs, we connect the **S pin** to Analog In (with A_numbers) on Open CR, and write `analogRead(pin_number)` in function `loop()`. DO NOT write `pinMode(pin_number,INPUT)` for an analog pin as it causes the analog pin to behave as a digital pin. (The analog pressure sensor code in the keyewiki website writes pinMode in `setup()`, which is wrong!!)

e. use `Serial.println()` to print results on serial monitor

f. Click the check mark on the Arduino IDE interface to compile code and the "-->" to push to the OpenCR board.

g. At this point, you should be able to see outputs from the serial monitor (click the magnifier on upper right to open the serial monitor interface).

5. Modify library files in OpenCR package on your local machine

   a. We will modify a sensor called **illumination** in the `Sensor_State` node (the same node where the bumper sensor is called).

   b. `Sensor_State` node is implemented in this file: `Library/Arduino15/packages/OpenCR/hardware/OpenCR/1.4.19/libraries/turtlebot3/src/turtlebot3/turtlebot3_sensor.cpp`

c. Based on your previous testing code for the sensor, modify these two function in the library file: `Turtlebot3Sensor::init()` is where you put your `setup()` function and replace the code for illumination sensor in `Turtlebot3Sensor::getIlluminationData(void)` with your sensor code in your `loop()` function

d. An example of modified sensor state file for digital push button sensor: https://github.com/lwu5/recyclable_trash_sorter/blob/main/turtlebot3_sensor.cpp

6. Push to the OpenCR board using the “-->” on the Arduino IDE interface.

7. Disconnect OpenCR board from your local machine and connect the USB port back to Pi.

8. Install TurtleBot3 via Debian Packages on your VM:

```
$ sudo apt-get install ros-noetic-dynamixel-sdk
$ sudo apt-get install ros-noetic-turtlebot3-msgs
$ sudo apt-get install ros-noetic-turtlebot3
```

9. Test the sensor node by running this command on your VM:

```
$ rostopic echo sensor_state, read Illumination
```

10. DO NOT rerun the script on the course website:

    a. DO NOT run commands under “update firmware and files”: https://classes.cs.uchicago.edu/archive/2022/spring/20600-1/turtlebot_assembly_setup.html

    b. It will cause overwriting the arduino script we just modified and pushed to OpenCR board (Undo everything we did)