mBot Resources and Tips for Your CG1111 Project

1. Assemble your mBot according to the instructions given inside the box. Note that you do not need to install the Bluetooth module; please safekeep it.

2. You can download the latest Makeblock libraries from the following link:
   https://github.com/Makeblock-official/Makeblock-Libraries
   - Follow the instructions on the webpage to add the libraries into the Arduino IDE on your laptop.
   - You can find the source code for the mBot's **factory firmware** by navigating to `/examples/Firmware_For_mBlock/mbot_factory_firmware`
   - You can learn a lot about how the mBot works by going through the factory firmware, and observing the mBot's behavior when it runs the firmware. You can learn about:
     - How to control the built-in RGB LEDs on the mCore Main Control Board (mBot's brain)
     - How to control the speed and direction of the left and right wheel motors
     - How to detect a black line underneath the mBot
     - How to play a certain sound frequency through its on-board buzzer (speaker)
     - How to use its ultrasonic sensor for ranging
     - Etc.
   - The mBot has many other compatible peripheral devices, which are not included with the basic kit you have. Be careful not to use the wrong code for a non-existent peripheral device (commonly-made mistake).

3. The schematic diagram for your mCore can be downloaded from this link:
   https://github.com/Makeblock-official/mBot_Firmata/blob/master/hardware/mCore.pdf
   It serves as a very good reference for understanding how the firmware works with many of the components on the mBot.
   (Hint: From the schematic diagram, you can also figure out what is the analog pin number from which you can read the voltage of the on-board "Light Sensor", which is similar to the LDR seen in Week 10 Studio 1.)

4. As seen in Figure 1, the mCore has four RJ25 ports (for connecting cables that look like telephone cables). These ports can be used for connecting to other mBot accessories/sensors. From the schematic diagram of the mCore, you can see that only Ports 3 and 4 support reading analog voltage values (pins A0, A1, A2, and A3). This means that you can only use one of these two ports to read the analog voltages of your infrared proximity sensors (refer to Week 10 Studio 1).

5. The RJ25 adapter provided to you (see Figure 2) has two sets of pins allowing breakout connections to two customized sensors. You will be using it to connect to two sets of self-built infrared proximity sensors that allow your mBot to sense its distances from the left and right maze walls. As seen in Figure 3, each of the two sets of 3-pins consists of a GND (i.e., ground) pin, a VCC (i.e., 5 V) pin, and a sensing pin (S1 or S2). The two sensing pins, S1 and S2, are mapped to pins 5 and 6 inside the RJ25 port. Refer to the mCore schematic diagram

to find out which analog pin you should read to obtain the voltage for each of the two self-built infrared proximity sensors.

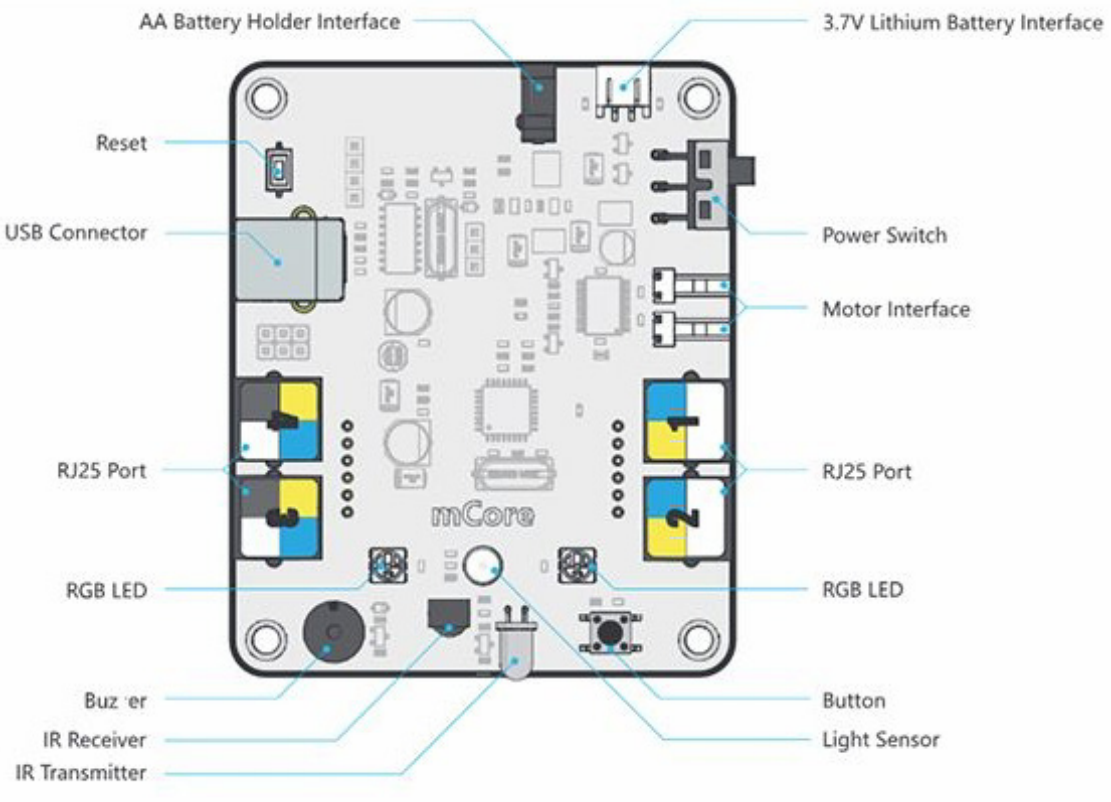

Figure 1: Layout of the mCore Main Control Board

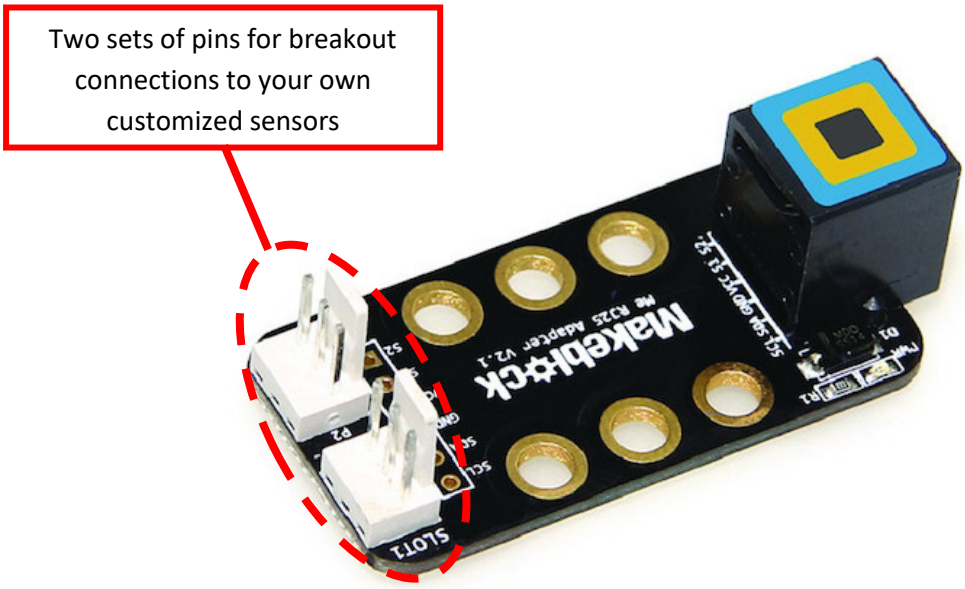

Figure 2: Top view of the RJ25 adapter

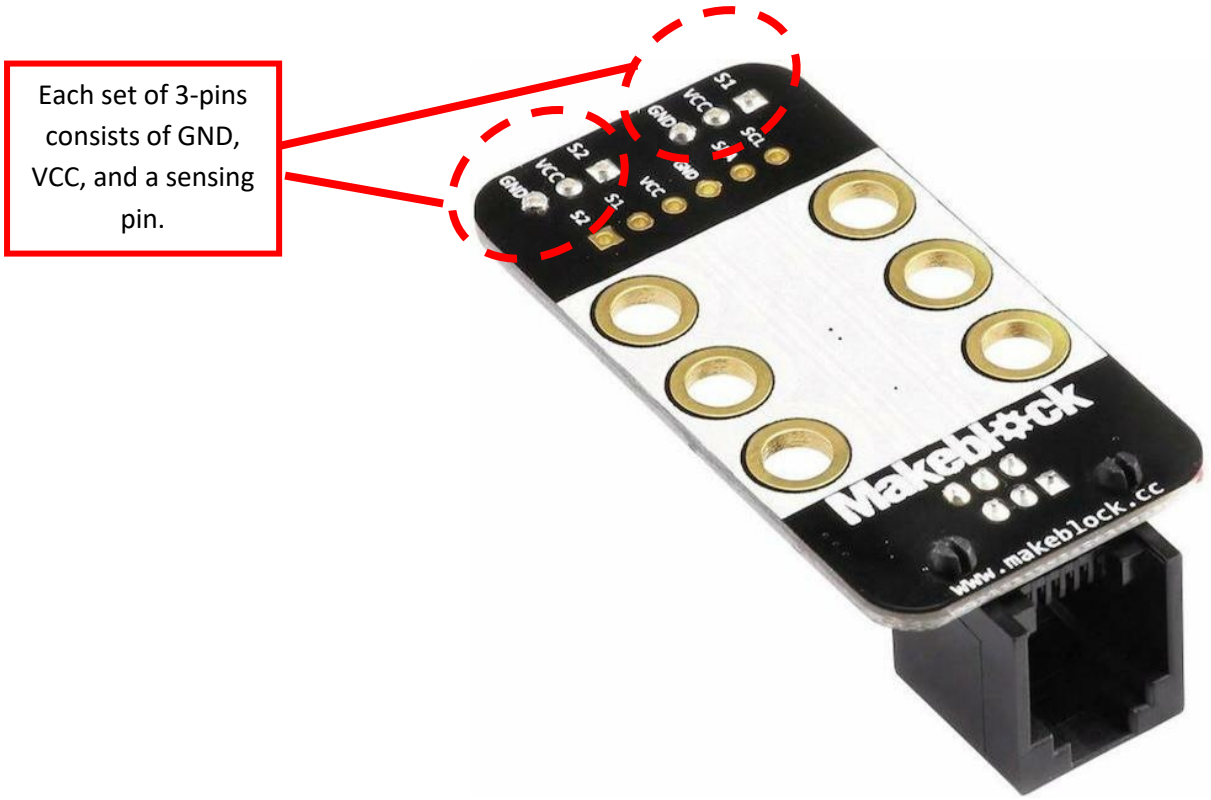

Figure 3: Bottom view of the RJ25 adapter

6. To effectively use the RGB LEDs and Light Sensor on the mCore for colour detection, you need to remove the top of the mCore case. You should place back the top of the mCore case when transporting the mBot, so as to protect its sensitive electronics from any accidental damage (e.g., due to water spillage, electrostatic discharge, metal conductors coming into contact with the electronics and shorting them, etc.)

7. Your mBot is powered by four AA rechargeable batteries. To obtain consistent performance, always ensure that the batteries are sufficiently charged. You may want to measure the voltage between VCC and GND of your mBot (e.g., you can measure them at your RJ25 adapter) from time to time to ensure that it is still kept at 5 V. If the batteries are too weak, the voltage VCC will fall below 5 V, and this can lead to unintended outcomes (e.g., wheel motors spinning slower than usual, infrared proximity sensors deviating from previous calibrated values, etc.).

8. There is no way for your mBot to know how much it has turned, since the basic mBot does not come with a gyroscope. You need to be innovative about how you can accomplish 90 degree turns, 180 degree U-turns, etc. (e.g., through hardcoding the turning time through trial and error). You should now understand why your batteries' charge level is important.

9. Ensure that you route your mBot's wires/cables properly. They should not protrude out too much as they may get entangled to the maze walls. They must also remain intact when your mBot accidentally bumps into the walls. Very importantly, they must not scratch against your mBot's wheels; otherwise the rotational speed may become inconsistent.