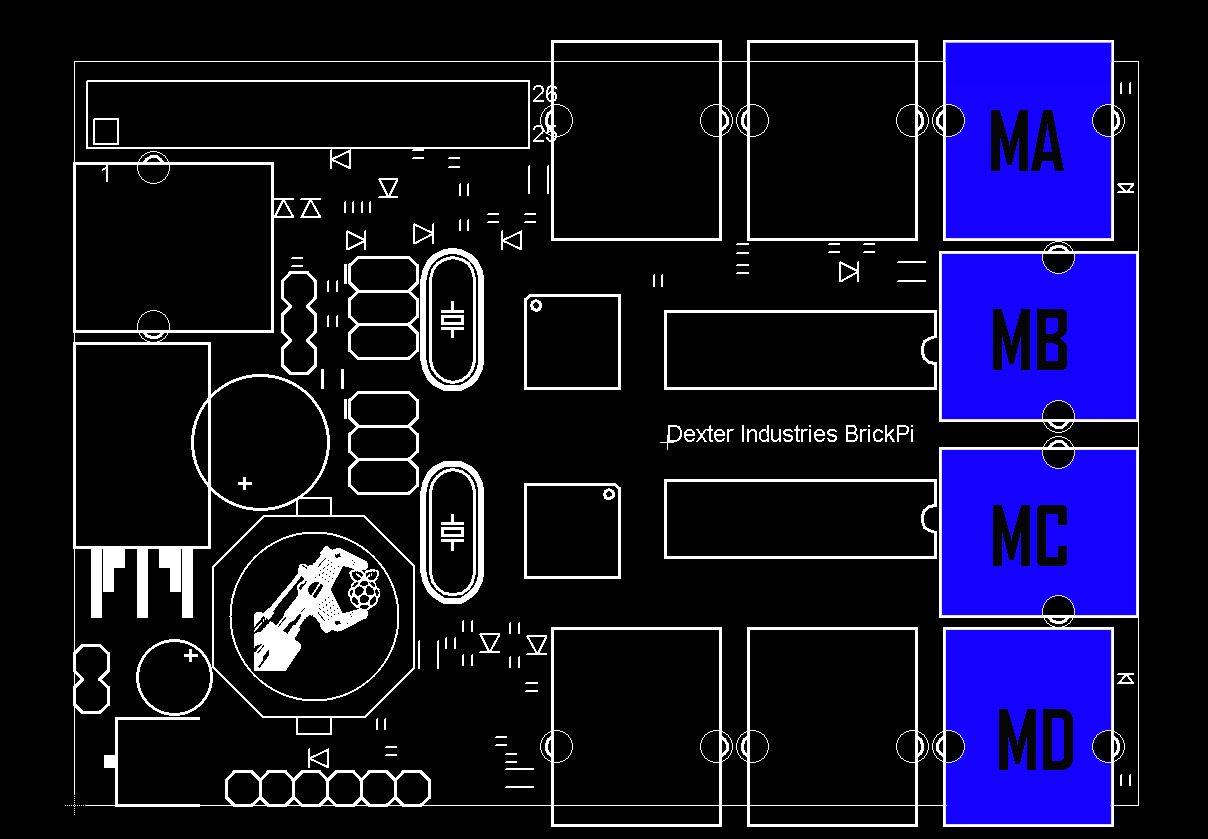
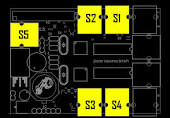
**Introduction To Using Motors And The BrickPi**

Now that you know how to write programs and navigate around the Linux OS on your raspi, it is time to start using BrickPi. BrickPi is a shield that attaches to the raspi, which enables the user to connect and communicate to Lego motors and sensors. In this Lab you will be learning about the basic information relating to the BrickPi and how to run Lego Motors on the BrickPi.

Introduction to BrickPi

 The BrickPi has 5 sensor and 4 motor ports for a total of 9 ports. In order for the Lego Motors to work they must be plugged into a motor port, likewise sensors must be plugged into senor ports. The diagrams below show the distribution of the ports.

The BrickPi also has the capabilities to run off a battery supply, enabling robots to run cordless. The plug for the power supply is located directly above the raspi micro usb power supply. Note: Do not program while on battery supply, this is just wasting batteries. Instead plug in the raspi to the micro USB charger while you are programing and only use the battery supply when you are running the robot. You can plug it in while on battery power, and then unplug it again when you want to go cordless.

The BrickPi case is also extremely helpful, it will allow for you to directly connect to Lego pieces. This makes integrating the BrickPi directly into the structure of your Robots extremely easy.

Running Lego Motors With BrickPi

Next you are going to learn how to actually get a Lego motor to run using python and the BrickPi. In order for you get some more practice navigating around in the terminal, you will need to go and find a program called *LEGO-Motor\_Test.py.* It should be stashed away in to the *Sensor\_Example* directory, which you found in the last lab. Once located, use *sudo nano* to take a look at the file. It should look like something below.

Next exit the program by hitting control x, and saying no to saving. Then plug two Lego motors into ports A & B, and run the program using < sudo python LEGO-Motor\_Test.py *>.* Once you have executed the command the two motors should be running with altering speeds. Now that you have successfully ran the program it is time to start dissecting the code to see how you can write some of your own.

**from BrickPi import “**function**” or “**\* **“:** This line of code is a must have in order to run the BrickPi and use any of it’s sensors / motors. It imports all the BrickPi functions from the directory BrickPi. For future reference you can import everything from a directory by just using a \*.

**BrickPiSetup() :** Another must have in order to use the BrickPi. This command creates the connection from the raspi to the BricPi. And it usually comes right after the imports.

**BrickPi.MotorEnable[PORT\_A] = 1 :** This command starts up the motor port. In order for the port to run any motors you must first enable it. You can start any port by just changing out the “A” for either: B, C, or D.

**BrickPiSetupSensors() :** Another must have command in order for the BrickPi to work. You **MUST** include it in every program using the BrickPi even those that do not use sensors. Also it needs to be after you have assigned the motors to their respective ports.

**BrickPi.MotorSpeed[PORT\_A] = 200 :** This command sets the speed of the motor plugged into port A to (for this example) 200. The range of values you can set the motors speed is -255 to 255. Negative values cause the motor to go in the opposite direction. Note: the motors will not change their speed after this command, just the value will change in the program but not the actual speed of the motor. One more command must be run in order to cause the physical change. You can treat this as a variable that can be written over in order to change the speeds to respond to a command.

**BrickPiUpdateValues() :** This command will update the physical motors to whatever value they have been assigned to in the program. It must be in a while loop or separate thread in order for the motors to continue to run. It must be constantly providing data to the motors for them to run if the data stream is interrupted the motors will stop running.

Now that you know all of the commands necessary to run Lego motors, it is time to write your own program.

**Lab 3.** Running 4 Motors!!!

Just as the description says, you are to write a program that runs four separate Lego motors all at different speeds. Please submit a copy of the program, with annotation explaining what each command does / your thought process.

Researching / writing your own code.

It is now time for you to start figuring things out for yourself. As any good engineering and especially as engineering students you must be able to research a topic find the useful information and then use it for your own purposes. Thus for your next two assignments you need to do just that. You are free ( encourage ) to use the internet, and especially < <https://github.com/DexterInd/BrickPi_Python> > which is the github library for BrickPi, to research commands and building techniques to complete the following task.

**Lab 4.** Put them up.

You are to build a Lego ball shooter that shoots balls at either semi automatic, or automatic, depending on what common the user inputs. The user also needs to be able to stop the gun. This project has two sides to it: the physical gun, which is mechanical and the software. So when writing up your lab report you need to include details about how you built the gun and also how your annotated program.

**Lab 5.** Exact distance.

Now it is time to get your BrickPi mobile. You are to build a car that will go two feet (two floor tiles) stop for a couple of seconds then go back 1 foot stop and then repeat. Then after it has progressed a total of ten feet it needs to rotate 180 degrees and return to the starting line where it will stop. It can go straight to the starting line on its return journey, it does not need to stop and go. This will require both hardware and software, so before lunching into the project take some time sketch out your design and outline the basic idea for your program. Remember to do some research about the software there may be a command that could make your life easier. Also make sure to write up your lab report explaining your hardware and your software, also include any calculations / measurements you make.