Experiment #1 Mapping the robots motors

Preparations. Go to www.arduino.cc and download and install the Arduino IDE. Launch the IDE and select Tools/Board. Set the board to Arduino/Genuino Mega or Mega 2560. Select Tools/Processor and select the ATMega2560.

For the first basic motion workshop experiment you will be given a robot with a unique wiring configuration and you task is to identify the specific motion bytes your program will use to control direction and motion.

The standard UNO Robotics club robot uses 4 NEMA 17 stepper motors with Mecanum wheels. The stepper motors are driven by a four channel motor controller board. The motor controller boards have common connections for the normal enabling pins and 4 sets of direction and stepping pins, one set for each motor. The enabling pins are hardwired to the robots VCC and GND. The 4 sets of direction and stepping pins are wired to PORTL, pins 42 – 49, of the Arduino Mega 2560 allowing simple programming to control speed and direction

The Mecanum wheels allow the robot to move in any direction. Initially we limit our study to the basic motions of forward, reverse, rotating left, rotating right, and moving sideways to the left or right. Later we will study some other hybrid motions for moving diagonally, moving in arcs, and pivoting.

The basic robot is controlled by an Arduino Mega 2560 microcontroller which has general purpose I/O pins, ports, A/D pins, and 5 16-bit counter/timers. PORTL is an 8 bit port which can be directly written or read. The direction of operation of the port must be set before the port is used. This is done with the command DDRL=B11111111;. The Arduino is connected to the controllers through header pins around the perimeter of the board. The header pins for PORTL are numbers 42 through 49 which are mapped to bits 0 through 7 of the PORTL byte. This allows all 8 controls signals to be changed in a single command. Due to wiring consideration we map the even number pins/bits 0,2,4,6 to the motion stepping pins of the motor controller and the odd number pins/bits 1,3,5,7 to the direction control pins.

The direction pins/bits directly tell the direction of motion of the wheels. The motion stepping pins tell the stepper motor to move one step each time the pin switches from 0 to 1. The direction bits are set before motion and the motion stepping bits are toggled repeatedly to make the robot actually move.

Launch the IDE and enter the following code into the setup function.

```
DDRL = B11111111;
PORTL = B00000000:
byte M1 = B01000000;
byte M2 = B00010000;
byte M3 = B00000100;
byte M4 = B00000001;

for (int I =0; I < 5000; i++) {
PORTL ^= M1;
delayMicroseconds(1000);
                                }

for (int I =0; I < 5000; i++) {
```

```
PORTL ^= M2;
delayMicroseconds(1000);
                        }

for (int I =0; I < 5000; i++) {
PORTL ^= M3;
delayMicroseconds(1000);
                        }

for (int I =0; I < 5000; i++) {
PORTL ^= M4;
delayMicroseconds(1000);
                        }

////// End of code
```

Select the compile button “checkmark” and edit the code until it compiles.

In Lab – Connect your computer to the robot with the provided USB cable. Select Tools/Ports and select the auto-detected port. Click on the download button “right arrow”. If the program downloads properly the robot is ready to run.

Place the robot on the provided blocks and verify the wheels are not touching the ground. Disconnect the USB cable and turn the robot on. The motors should turn one at a time. Record the sequence and direction of the motors motion in Table 2 of the worksheet at the end of this document.

Using Table 1 and Table 2 you must next determine the bit patterns for the six basic motions. To determine the direction bytes, you must identify the motors M1 through M4 and their natural direction. The natural direction is the direction the robot turns when its direction bit is set to “0”. These are the values you recorded in Table 2.

In a correctly wired robot the “0” direction bit tells the motor to turn clockwise which is forward on the right wheels and reverse on the left wheels. The direction byte ROTL = B00000000; sets all direction to cw for a rotate left motion. Unfortunately not all robots are not wired correctly so you must adjust the

Instructions to fill in Table 3.

The rows of Table 3 are in the same order as Table 1. Go through Table 3 column by column and row by row. For each row of table 3, go through the columns M1 through M4. Read the corresponding location from Table 2. Use the location from Table 2 to index into Table 1. Compare the natural direction of the motor to the desired direction in Table 1. If they are the same enter “00” in the box. If they are different enter “10” in the box. Repeat this process until Table 3 is complete.

Checking the direction bytes

In the Arduino IDE select File/New. In the setup function enter the following code replacing the “?”s with the patterns from Table 3.

```
/// Code start
DDRL = B11111111;
byte allMotors = B0101011;
byte fwd = B????????;
byte rev = B????????;
byte rotl = B????????;
byte rotr = B????????;
byte strl = B????????;
byte strr = B????????;

 PORTL = fwd;
for (int i= 0; I < 2000; i++} {
        PORTL ^= allMotors;
        delayMicroseconds(500);
                        }
/// End code
```

Compile the code and edit as necessary. Download the code to the robot and place the robot on the blocks. Turn on the robot and observe the motion of the wheels. When the wheels stop turn the robot off. If all motion is correct your can place the robot on the table and turn it on.

If the motion is correct you can test the other direction bytes by editing the line PORTL = fwd; and replacing the desired direction byte; rev, rotl, rotr, strl, or strr. Once all directions are tested your lab is complete

Table #1 – Wheel direction vs robot motion

		Wheels			
		FL	FR	BL	BR
Motion	FWD	FWD	FWD	FWD	FWD
	REV	REV	REV	REV	REV
	ROTL	REV	FWD	REV	FWD
	ROTR	FWD	REV	FWD	REV
	STRL	REV	FWD	FWD	REV
	STRR	FWD	REV	REV	FWD

Table #2 – Actual motor response

Conditions – PORTL = B00000000;
Direction – FWD/REV. Location FL, FR, Bl, or BR

Motors	M1	M2	M3	M4
Location				
Direction				

Table #3 – Motion direction bits

	M1	M2	M3	M4
FWD				
REV				
ROTL				
ROTR				
STRL				
STRR				

Legend

FL – front left
FR – front right
BL – back left
BR – back right

FWD – forward
REV – reverse
ROTL – rotate left
ROTR – rotate right
STRL – strafe left
STRR – strafe right

M1 – motor 1
M2 – motor 2
M3 – motor 3
M4 - motor 4