ALL TERRAIN ROBOT 1

Wilmer Arellano © 2013

The Client's Need

Background

- This project is based on a library previously developed to control the motors of a robotic atv.
- For this reason you will see programming references to object "atv" like:

```
#include <Atv.h>
// Create an instance of Atv
Atv atv(0);
```

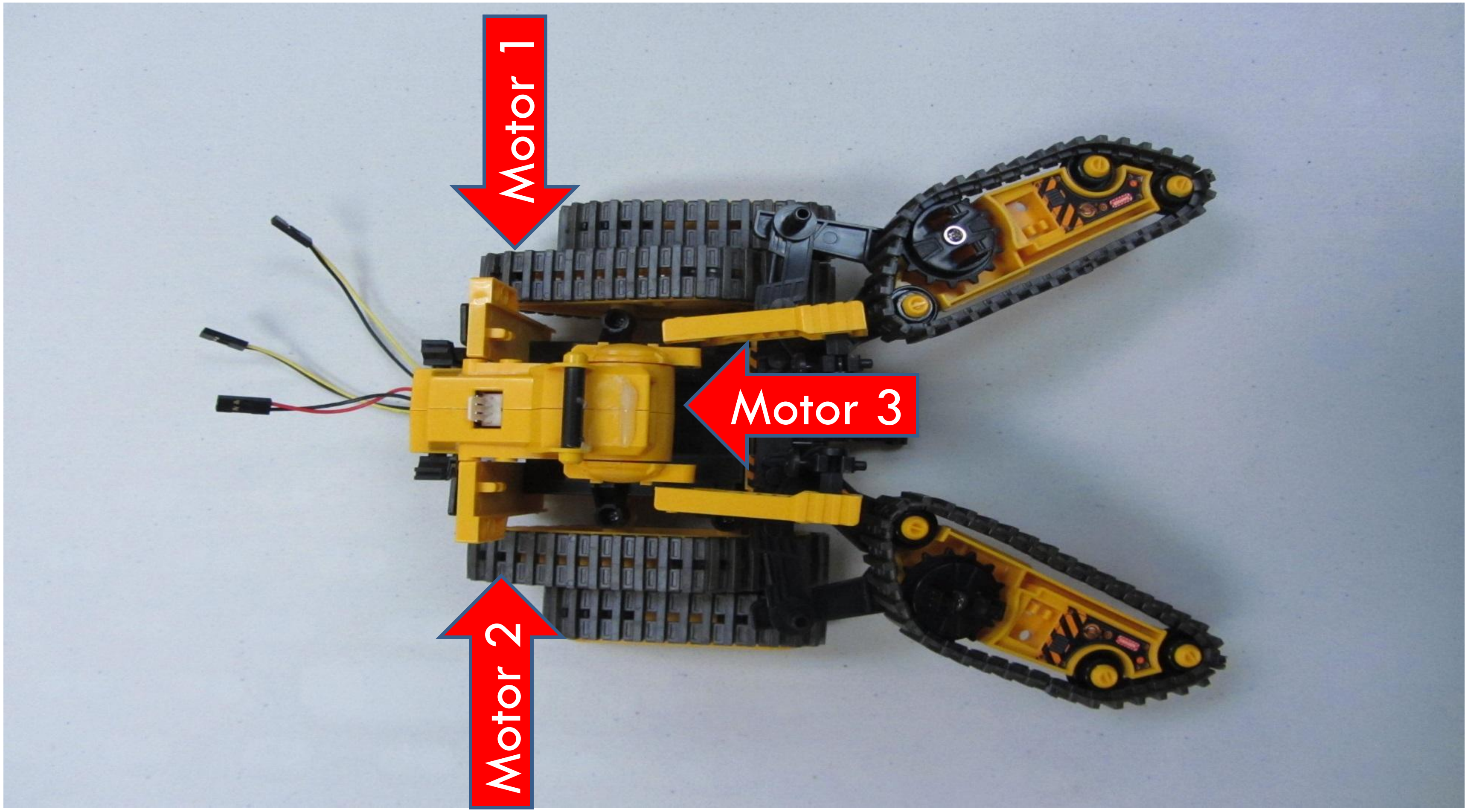
- And:

```
atv.moveMotor(1, LEFT, 5, 10);
```

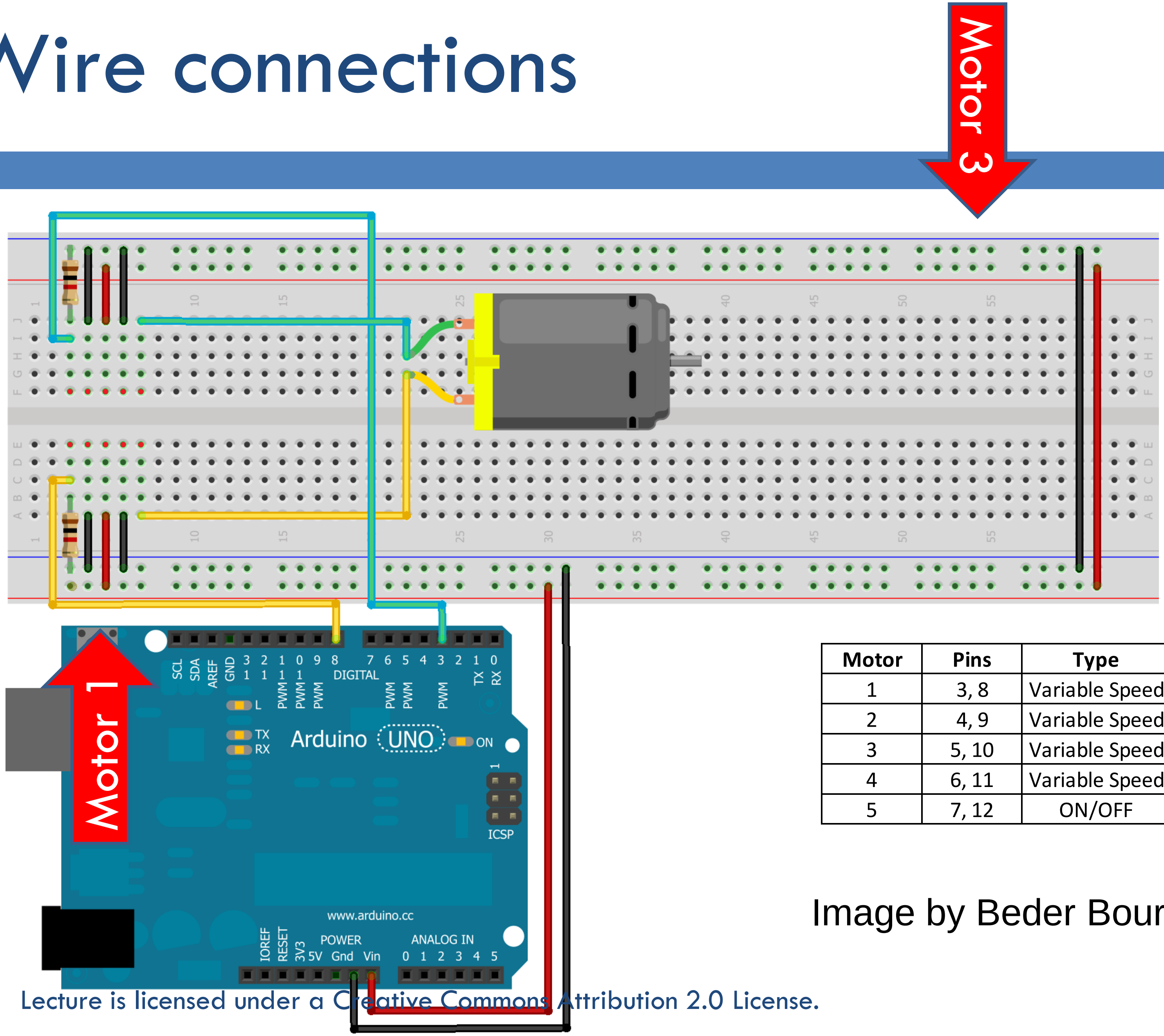
Hardware

- Next slides shows sensor connection to analog pin 0 and Motor 1 connection.

Hardware



Wire connections

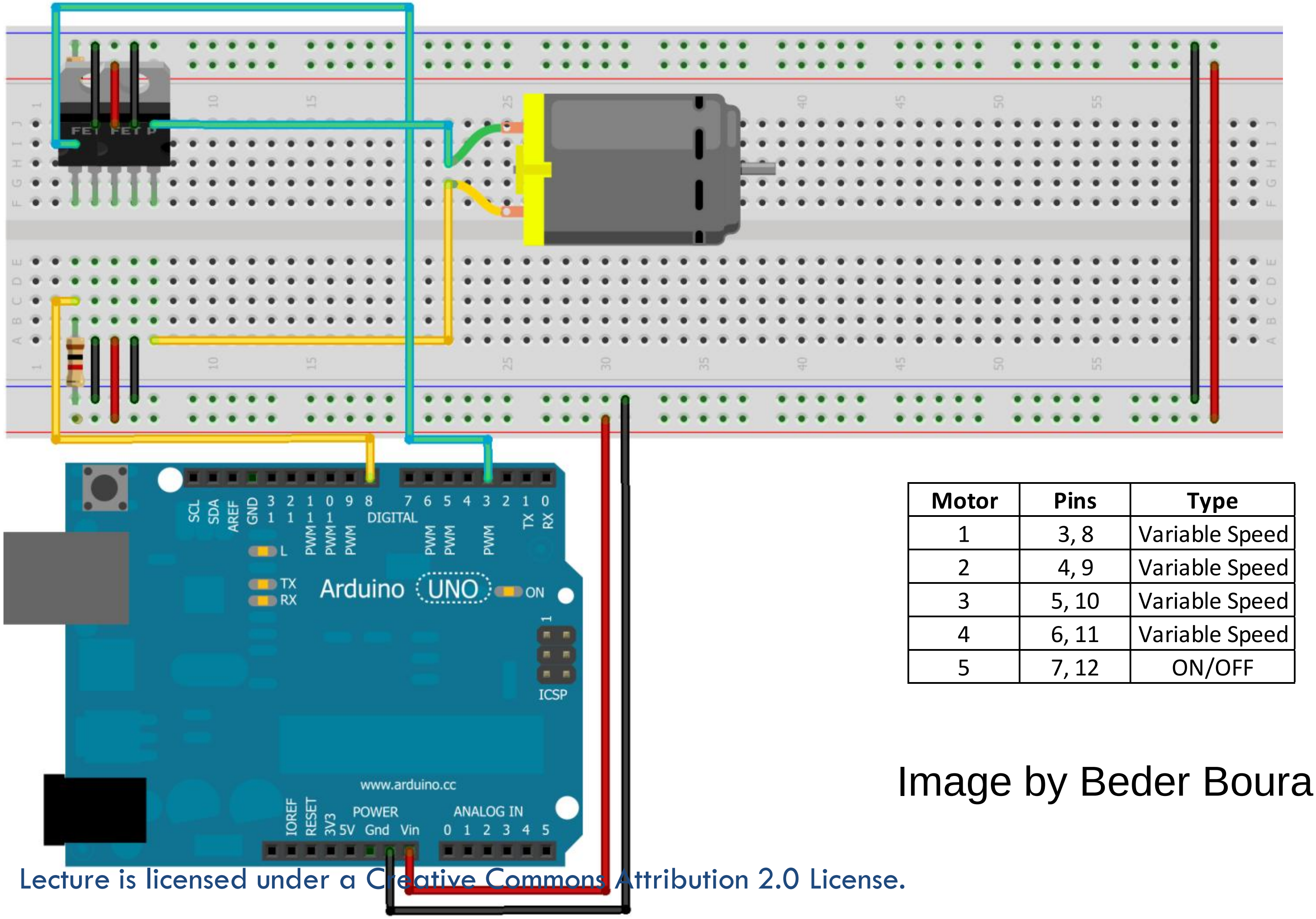


Motor	Pins	Type
1	3, 8	Variable Speed
2	4, 9	Variable Speed
3	5, 10	Variable Speed
4	6, 11	Variable Speed
5	7, 12	ON/OFF

Image by Beder Bourahmah

Made with Fritzing.org

Top half bridge connection

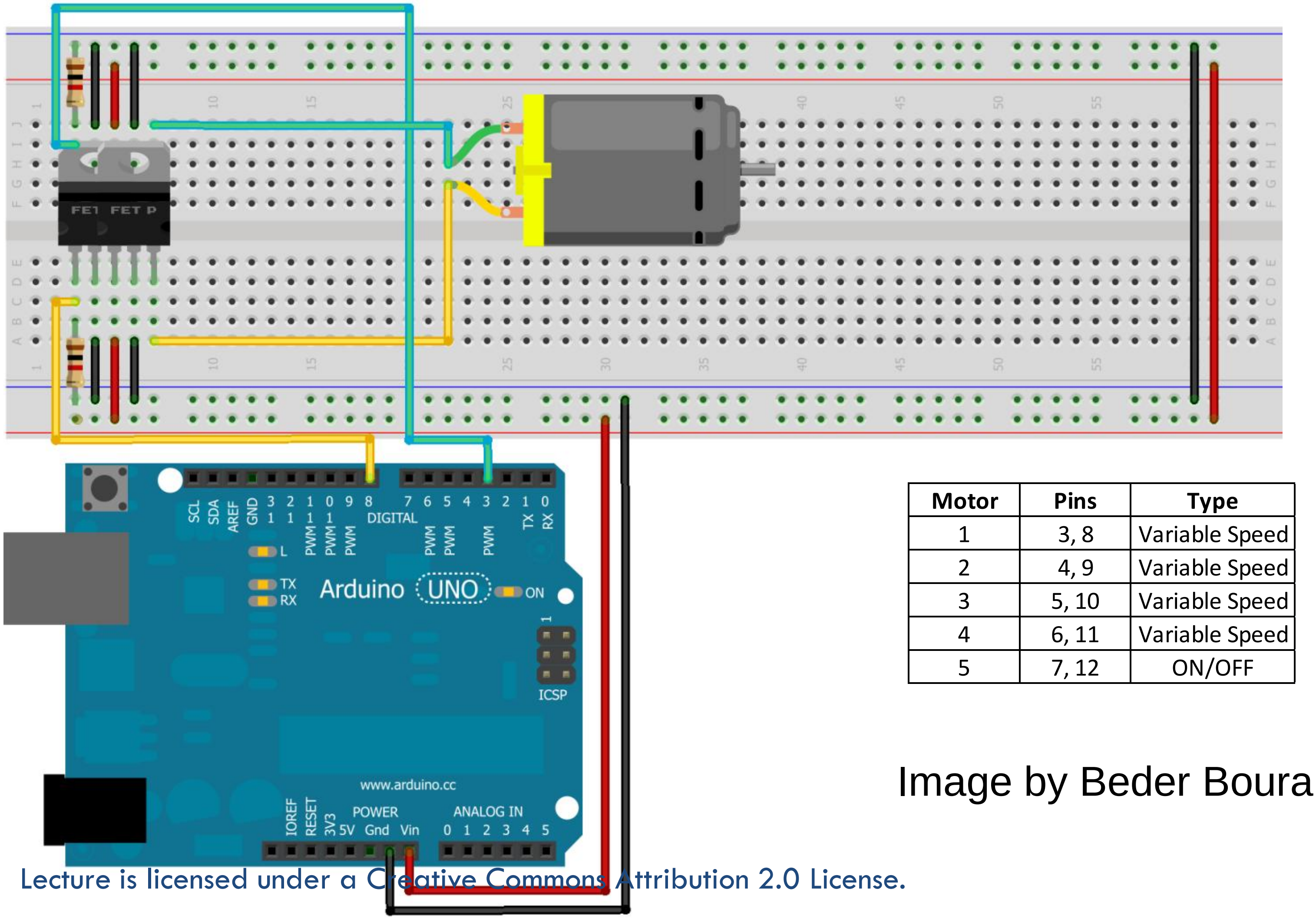


Motor	Pins	Type
1	3, 8	Variable Speed
2	4, 9	Variable Speed
3	5, 10	Variable Speed
4	6, 11	Variable Speed
5	7, 12	ON/OFF

Image by Beder Bourahmah

Made with Fritzing.org

Bottom half bridge connection

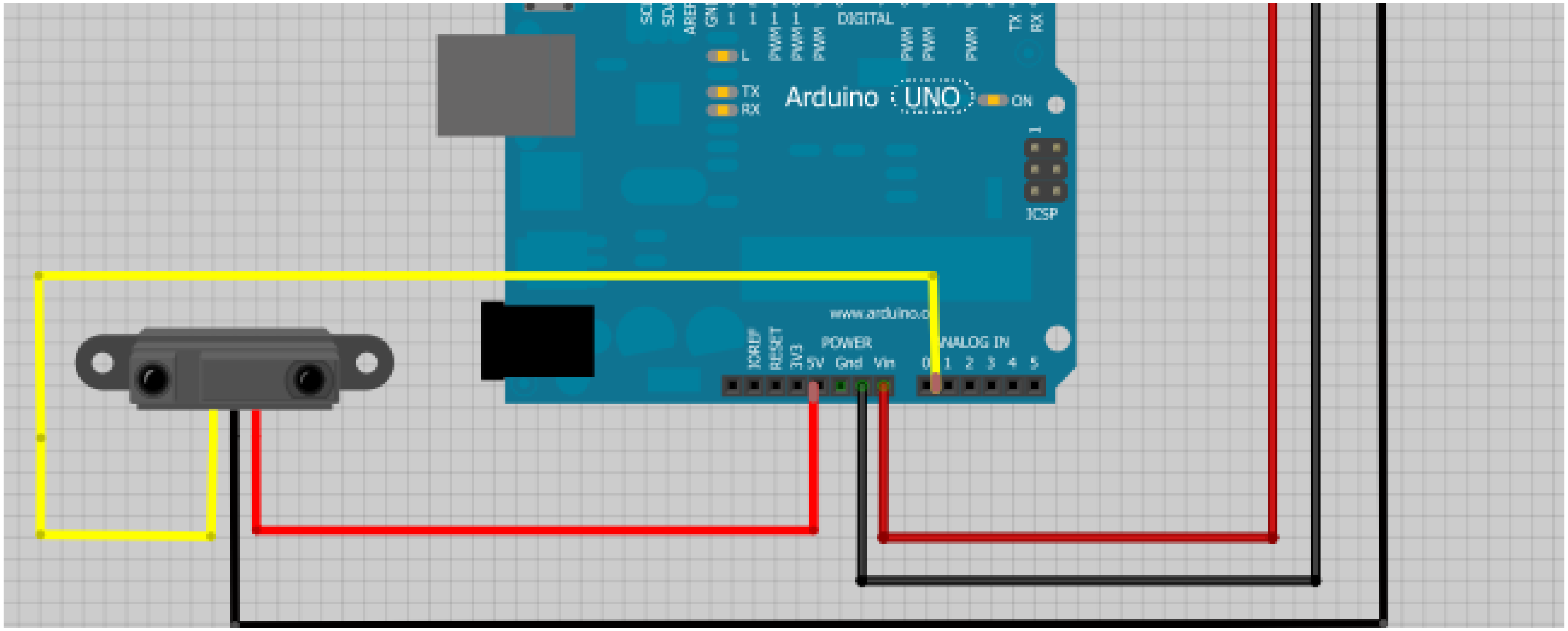


Motor	Pins	Type
1	3, 8	Variable Speed
2	4, 9	Variable Speed
3	5, 10	Variable Speed
4	6, 11	Variable Speed
5	7, 12	ON/OFF

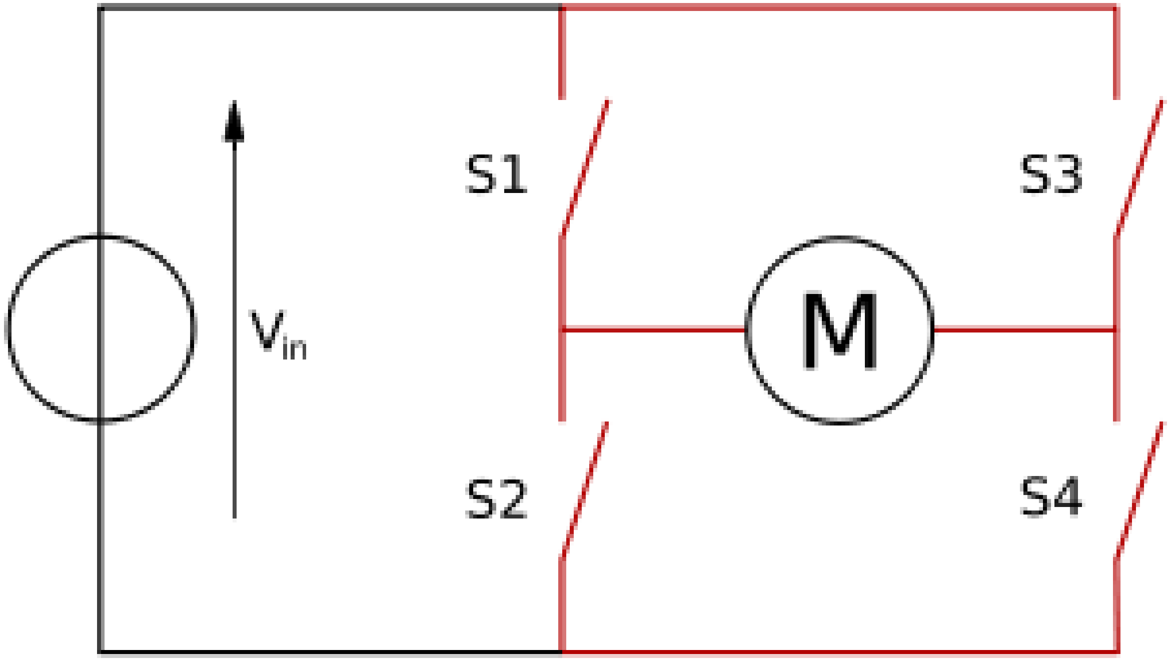
Image by Beder Bourahmah

Made with Fritzing.org

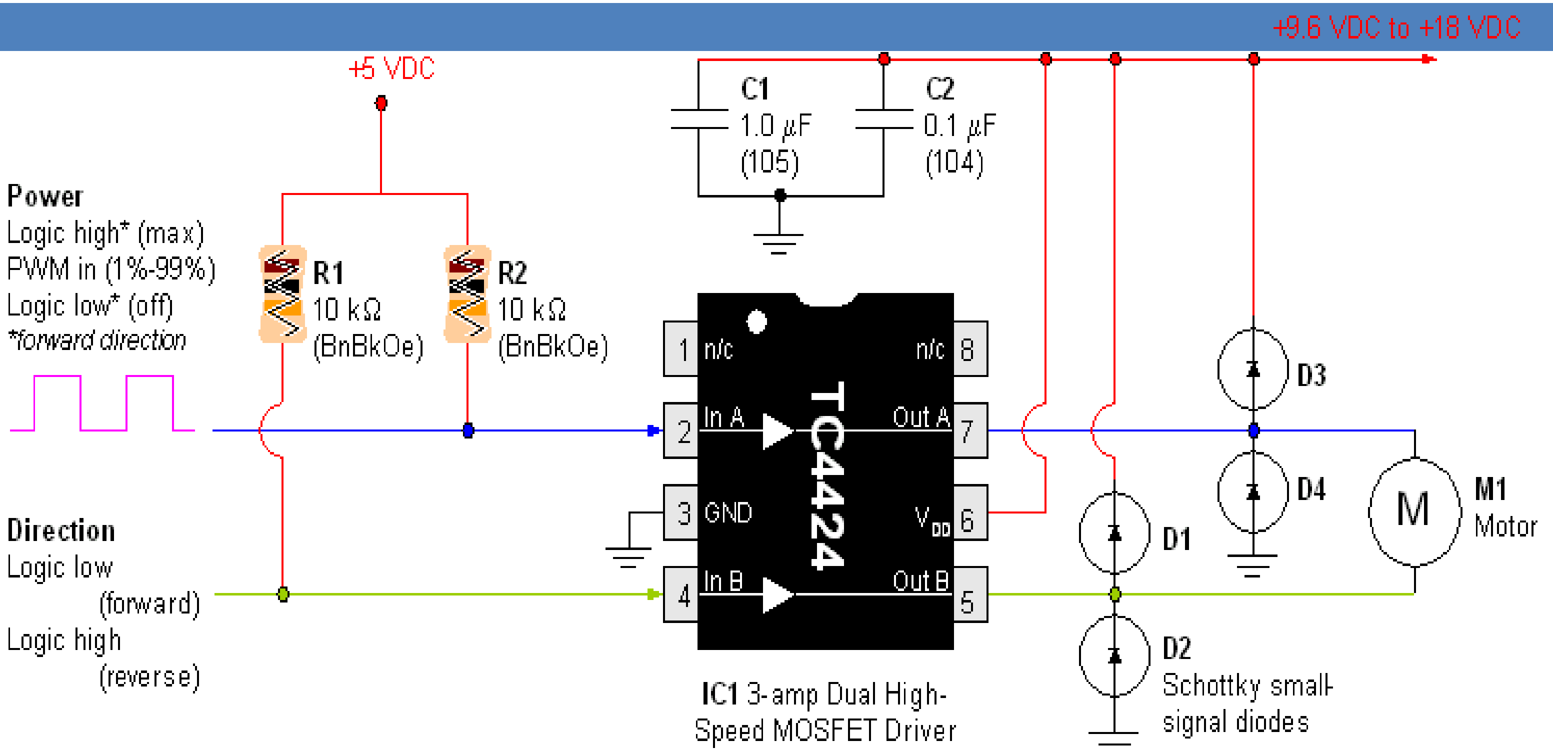
Sensor Connection



H-Bridge



H-Bridge



This is just an example, we prefer to use the TC4422 in the T0-220 package which has higher Current capability but is single channel. You can order free samples from microchip

Precautions

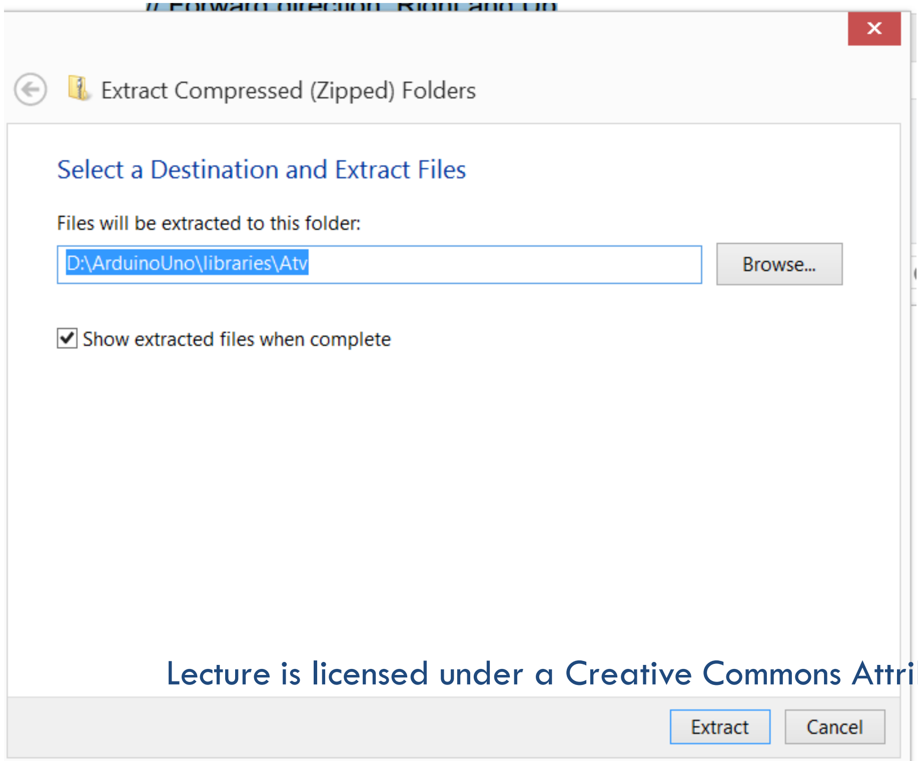
- The All Terrain Robot is a delicate device that may be damaged if operated beyond the mechanic limits
- If while you program the All Terrain Robot operation you observe a motion behavior that may compromise the robot integrity disconnect power immediately. Center the All Terrain Robot with the manual remote and make any corrections necessary to your program

Installation

- Download Library from:
  - http://web.eng.fiu.edu/~arellano/1002/Microcontroller/Atv.zip
- Unzip library and drop it in:
  - The libraries folder of your arduino installation. In my case:
  - C:\arduino-1.0.1-windows\arduino-1.0.1\libraries

Under Windows

Right click the file, extract all, remove Atv from destination



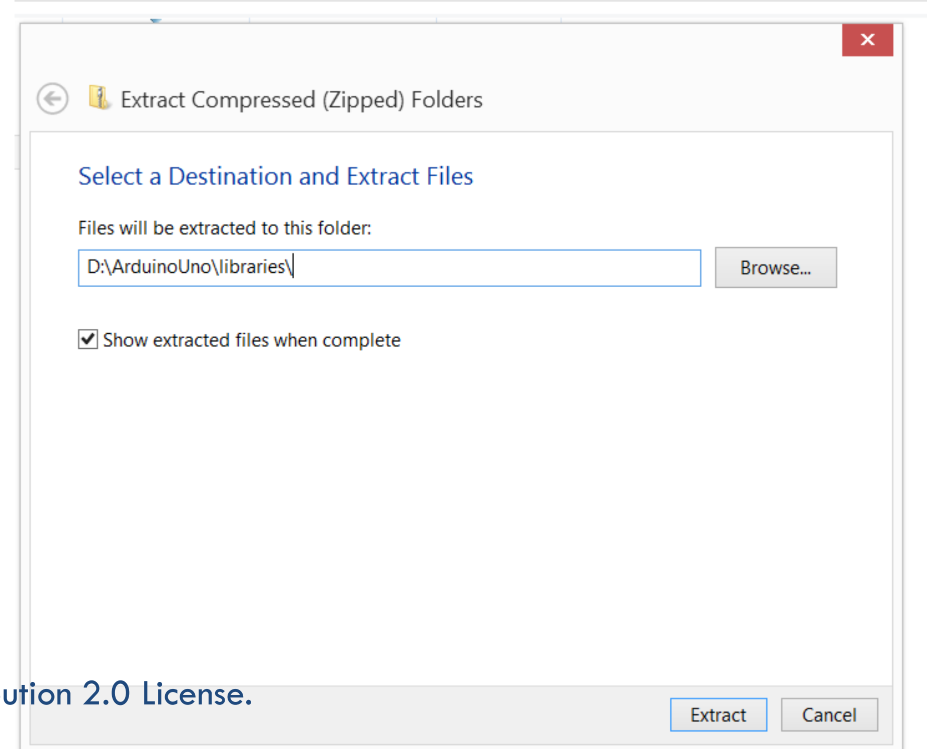
Extract Compressed (Zipped) Folders

Select a Destination and Extract Files

Files will be extracted to this folder:

D:\ArduinoUno\libraries\Atv

☑ Show extracted files when complete



Extract Compressed (Zipped) Folders

Select a Destination and Extract Files

Files will be extracted to this folder:

D:\ArduinoUno\libraries\

☑ Show extracted files when complete

- Include in your final report:
  - An explanation of a gear box operation. Particularize for the gear boxes you are using
  - How to measure power in a DC motor. Include several power measurements on the All Terrain Robot’s motors under different operating conditions

```
#include <Atv.h>
// Create an instance of Atv
Atv atv(0);
// Backward direction, Left and Down
int LEFT = 0, DOWN = 0;
// Forward direction, Right and Up
int RIGHT = 1, UP = 1;
int control = 1, temp;

void setup() {
 Serial.begin(9600);
 Serial.println("Hello");
}

void loop() {
  atv.checkData();
  while(control > 0){
    atv.moveMotor(1, LEFT, 5, 10);
    atv.moveMotor(1, RIGHT, 5, 10);
    control = control - 1;
    temp = atv.distance();
    Serial.print("Distance: ");
    Serial.println(temp);
  }
}
```

Constructor

- Include the library:
  - #include <Atv.h>
- Create an instance of the class:
  - Atv atv(int);
  - “int” tells the library where the distance sensor is connected.
- #include <Atv.h>
- // Create an instance of Atv
- Atv atv(0);

Create Mnemonics

- // Backward direction, Left and Down
- int LEFT = 0, DOWN = 0;
- // Forward direction, Right and Up
- int RIGHT = 1, UP = 1;

Initialize

```
int control = 1, temp;

void setup() {
 Serial.begin(9600);
 Serial.println("Hello");
}
```

Execute

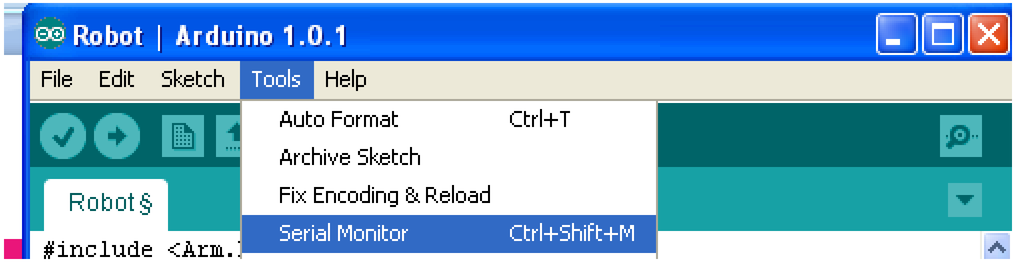
```
void loop() {
  atv.checkData();
  while(control > 0){
    atv.moveMotor(1, LEFT, 5, 10);
    atv.moveMotor(1, RIGHT, 5, 10);
    control = control - 1;
    temp = atv.distance();
    Serial.print("Distance: ");
    Serial.println(temp);
  }
}
```

Motor Control 1

- atv.moveMotor(int motorID, int Direction, int Speed, int time)
- Motor ID = 1, 2, 3, 4, 5
- Direction = RIGHT, LLEFT
- $0 < \text{Speed} < 11$
- $0 \leq \text{time} \leq 40$
  - Time = 40 will produce continuous motor motion regardless of the value of Speed
  - time = 0 will stop a motor in continuous motion
  - You may need to invert your motor connection for proper operation

Motor Control 2

- □ atv.checkData()
- □ This function with no arguments checks for serial data from the computer
- □ Use the serial monitor of the Arduino IDE to control the atv
- □ Type one or more commands and hit enter



```
#include <Arm.
```

Motor Control 2

- Commands
- Selection
  - a or A moves motor towards left
- Motion
  - s or S moves motor towards right
  - w or W moves motor up
  - z or Z moves motor down
  - a, z and s, w are interchangeable

```
void loop() {
  atv.checkData();
  while(control > 0){
    atv.moveMotor(1, LEFT, 5, 10);
    atv.moveMotor(1, RIGHT, 5, 10);
    control = control - 1;
    temp = atv.distance();
    Serial.print("Distance: ");
    Serial.println(temp);
  }
}
```

- A single motion command will produce a small movement a sequence of several motion commands of the same type will produce an ampler motion
  - After first execution only atv.checkData() will be executed continuously

Distance measurement 1

- temp = atv.distance();
- When this function is called an integer is returned with approximate distance between the sensor and an object
- Limitations
  - Max distance 80 cm
  - Object with a distance less than 10 cm will appear to be farther away

Distance measurement 2

- atv.checkData()
- This function can also be used to read distance
- Command
  - d or D Returns Distance

```
void loop() {
  atv.checkData();
  while(control > 0){
    atv.moveMotor(1, LEFT, 5, 10);
    atv.moveMotor(1, RIGHT, 5, 10);
    control = control - 1;
    temp = atv.distance();
    Serial.print("Distance: ");
    Serial.println(temp);
  }
}
```

Additional Experiments

1. Make robot move forward
2. Make robot move backward
3. Make robot spin
4. Mound distance sensor and test it
    1. You may need to use for or while loops

Functions

- Functions are segments of code that are placed in a different location other than locations void setup() { } or void loop() { }
- For example:

```
void spin(){
// Your code
}
```

- Void indicates that the function does not return a calculated value

Functions

- For example:

```
void spin(){
// Your code
}
```

- When you want to execute function spin() you do a call to the function:

```
spin();
```

Functions

- Next we will modify the original code to:
  - Include functions
  - Add some delay to manually position the griper
- Lines highlighted in red have change please read the comments.

```
#include <Atv.h>
Atv atv(0); // Creates an instance of Atv
int LEFT = 0, DOWN = 0; // Backward direction, Left and Down
int RIGHT = 1, UP = 1; // Forward direction, Right and Up
int temp;

void setup() { // We use this loop only as we want a single execution of the program
 Serial.begin(9600);
 Serial.println("Hello");
 atv.initialize(20); // The argument Determines how many second you have to manually position the ATV
 spin(); // "spin()" is a function call it will execute the code between the braces of "void spin()"
 // Program will return here after executing the code of spin()
}

void loop() {
}

void spin(){
  temp = atv.distance();
  while(temp > 50){
    // Replace the next comment lines with appropriated code
    // Move left motor forward
    // Move right motor backwards
    temp = atv.distance();
    Serial.print("Distance: "); // This line is optional if you want to monitor distance in the computer
    Serial.println(temp); // This line is optional if you want to monitor distance in the computer
  }
}
```

```
#include <Atv.h>
Atv atv(0); // Creates an instance of Atv
int LEFT = 0, DOWN = 0; // Backward direction, Left and Down
int RIGHT = 1, UP = 1; // Forward direction, Right and Up
int temp;

void setup() { // We use this loop only as we want a single execution of the program
  Serial.begin(9600);
  Serial.println("Hello");
  atv.initialize(20); // The argument Determines how many second you have to manually position the ATV
  spin(); // "spin()" is a function call it will execute the code between the braces of "void spin()"
  // Program will return here after executing the code of spin()
  // add more functions, like
  forward();
}

void loop() {
}

void spin(){
  temp = atv.distance();
  while(temp > 50){
    // Replace the next comment lines with appropriated code
    // Move left motor forward
    // Move right motor backwards
    temp = atv.distance();
    Serial.print("Distance: "); // This line is optional if you want to monitor distance in the computer
    Serial.println(temp); // This line is optional if you want to monitor distance in the computer
  }
}

void forward(){
  temp = atv.distance();
  while(temp > 17){
    // Replace the next comment lines with appropriated code
    // Move left motor forward
    // Move right motor forward
    temp = atv.distance();
    Serial.print("Distance: "); // This line is optional if you want to monitor distance in the computer
    Serial.println(temp); // This line is optional if you want to monitor distance in the computer
  }
}
```