01^{TM}SuperModified – Testing with Arduino
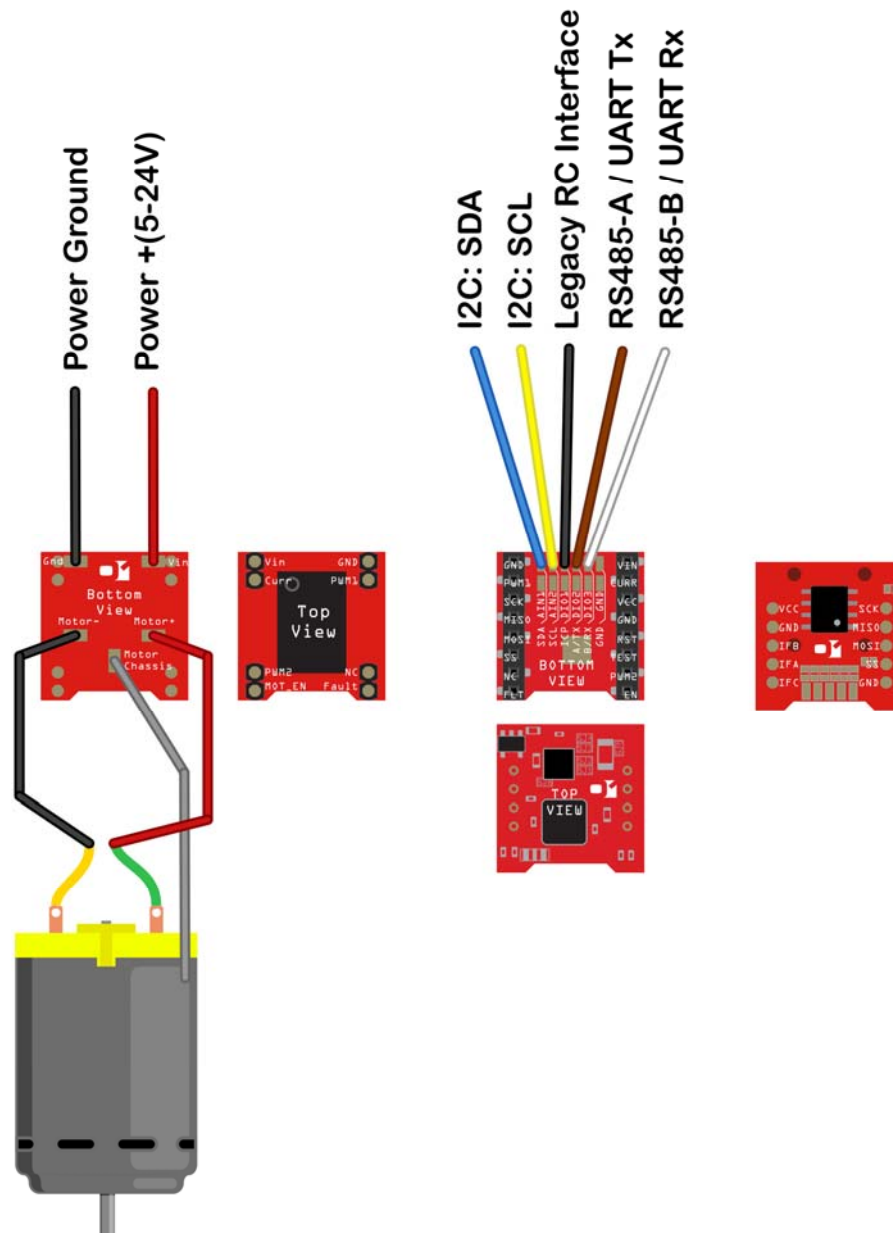
1. Preinstalled cables

The Supermodified controller comes with several wires soldered. By default the power and RS485 cables are soldered. I2C and legacy RC servo interface cables can also be soldered upon request. These are represented in the schematic below. Please note that the pre-soldered cables always have the illustrated colors.

ZERO ONE
MECHATRONICS

01™SuperModified

Miniature Controller for DC Motors

"*The robotic rebirth of the hobby servo*"

Testing with Arduino

2. Prerequisites

- **Windows XP or later.**

- **Arduinoinstallation. It can be downloaded at:** http://arduino.cc/en/main/software

- **01^{TM}SupermodifiedArduino library files.They can be downloaded at**http://www.01mechatronics.com/sites/default/files/docs/ZerooneSupermodified.zip

After downloading the archive with the SupermodifiedArduino library simply unzip it inside your Arduino libraries folder. This is usually C:/Program Files/Arduino/libraries.
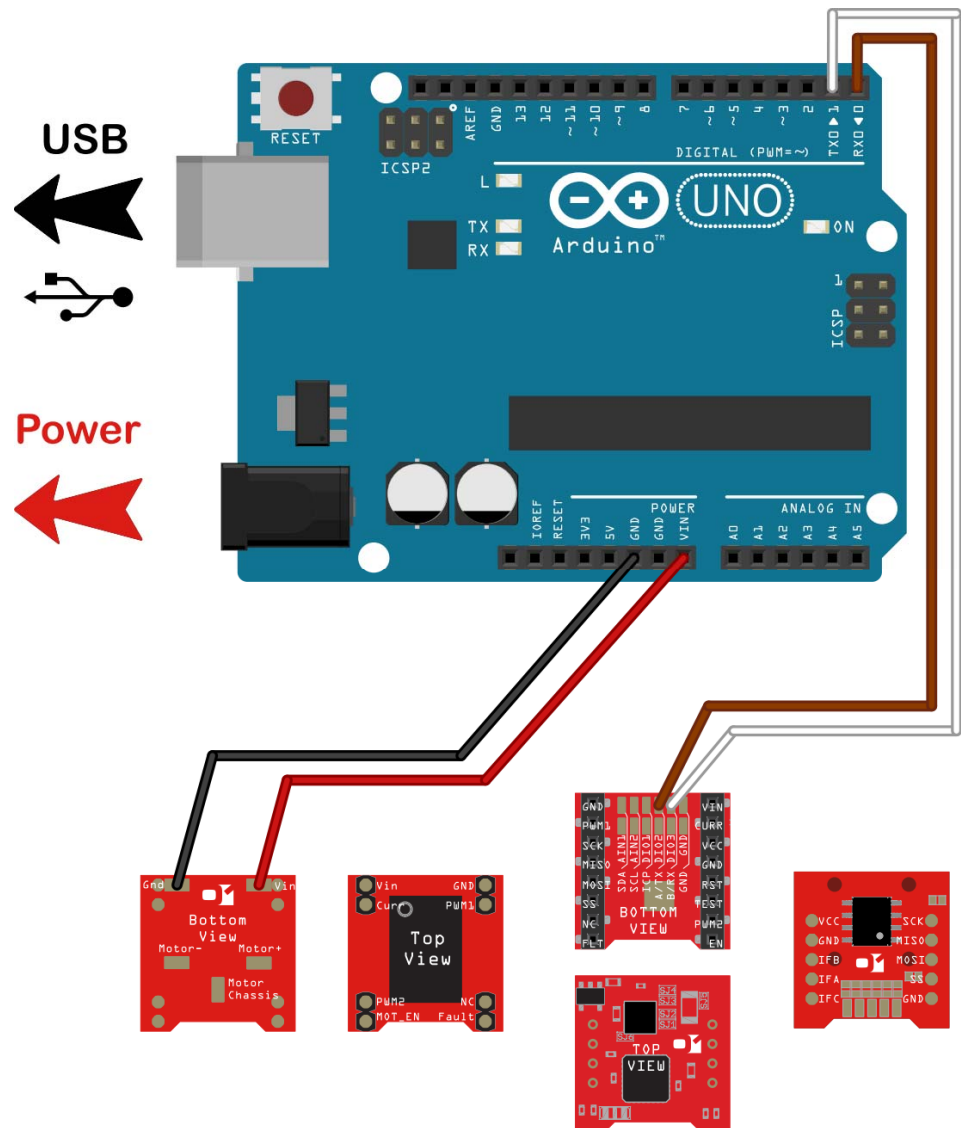
3. Connections

The Arduino library supports UART, RS485 and I2C communication to the Supermodified controller. Connections and examples for each case follow.

3.1. UART

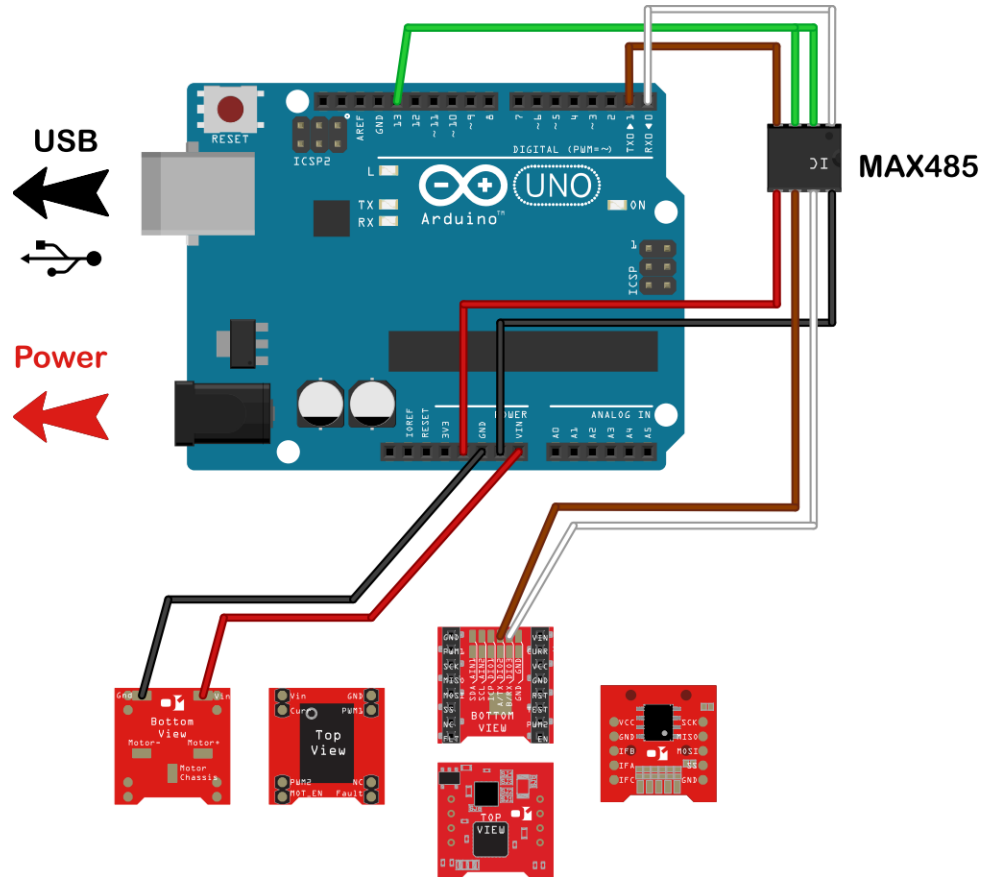The connections needed for UART operation are illustrated below:

Important Note: The Arduino**must** have the external power supply connected. Almost any motor can draw more than 500mA of current which is usually the USB maximum current.

3.2. RS485

The connections needed for RS485 operation are illustrated below. An external RS485 transceiver is necessary in this case as can be seen on the schematic.
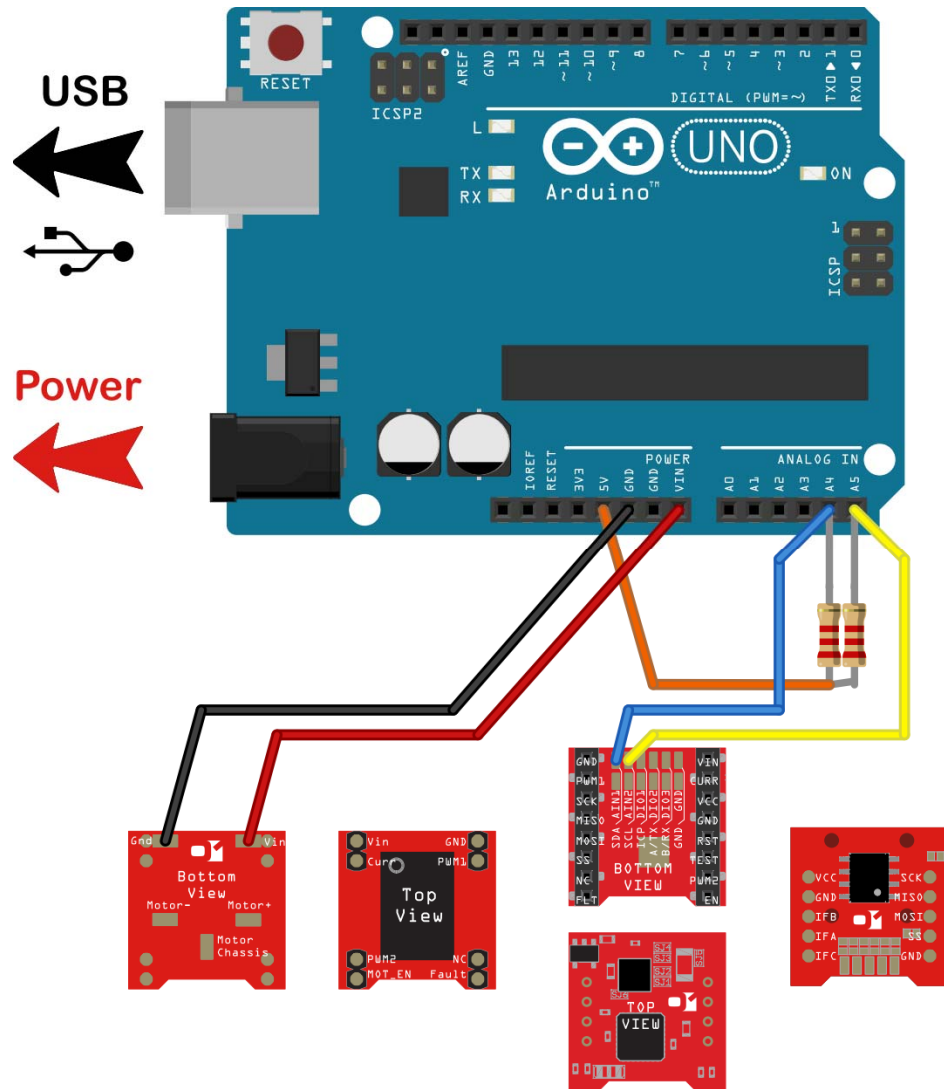
Important Note: The Arduino **must** have the external power supply connected. Almost any motor can draw more than 500mA of current which is usually the USB maximum current.

3.3. I2C

The connections needed for I2C operation are illustrated below. Although the Arduino Wire library activates the MCU internal pull up resistors, it is recommended to connect two external ones of approx. 2KOhm each.

Important Note: The Arduino **must** have the external power supply connected. Almost any motor can draw more than 500mA of current which is usually the USB maximum current.

3.4. Legacy RC interface

The connections needed for legacy RC interface are illustrated below. Note that according to 01^{TM}Supermodified firmware, standard PPM pulses can be interpreted in different manner, allowing e.g. for speed instead of position control. Also the angular range can be manipulated with different firmware. Please let us know your requirements and a custom firmware will be available as soon as you ask.

Important Note: The Arduino**must** have the external power supply connected. Almost any motor can draw more than 500mA of current which is usually the USB maximum current.
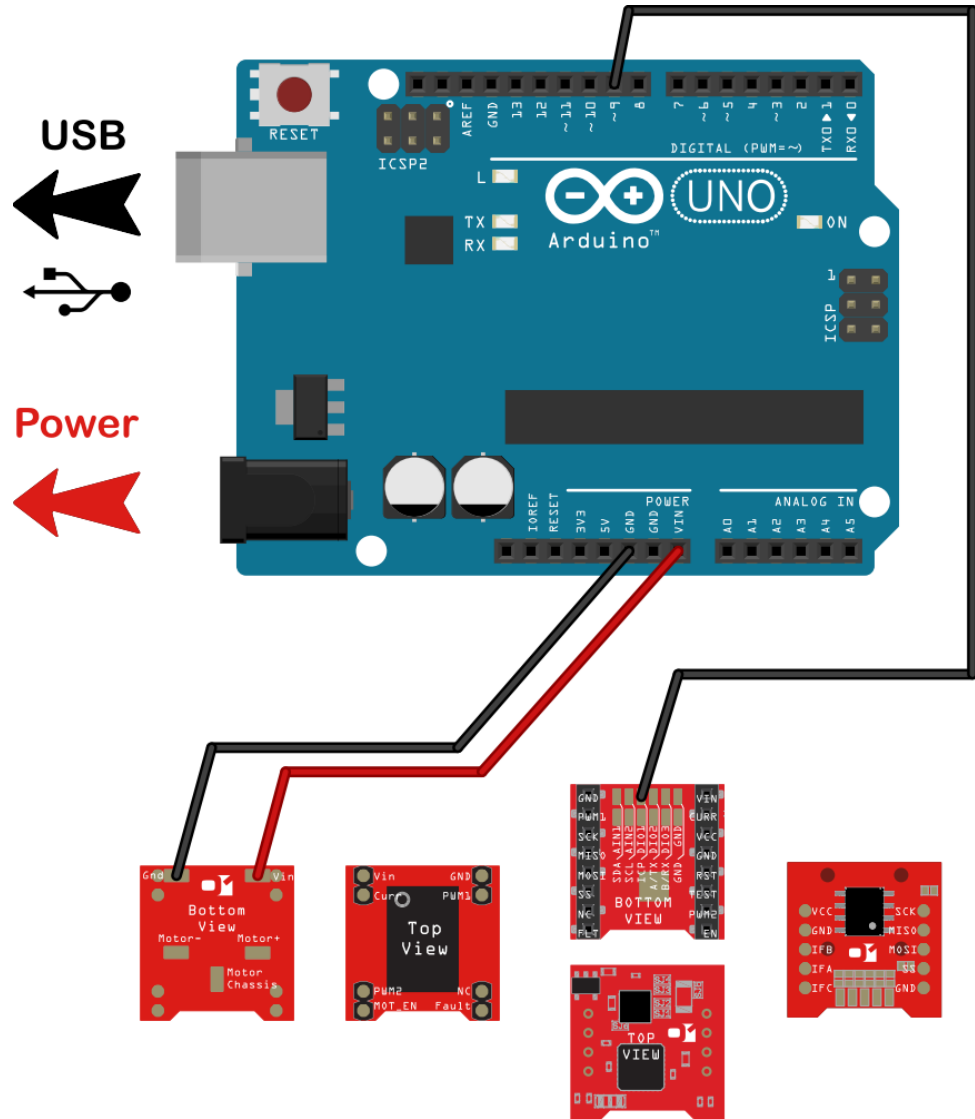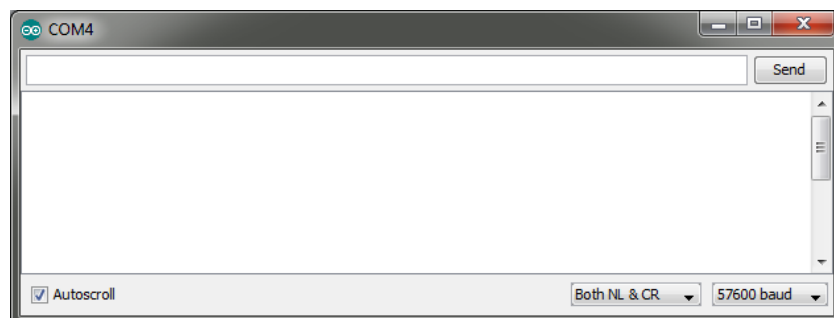
4. Testing with Arduino

After installing the library and making the necessary connections go ahead and try running the appropriate example.

From the Arduino menu go File > Examples >ZerooneSupermodified> test.
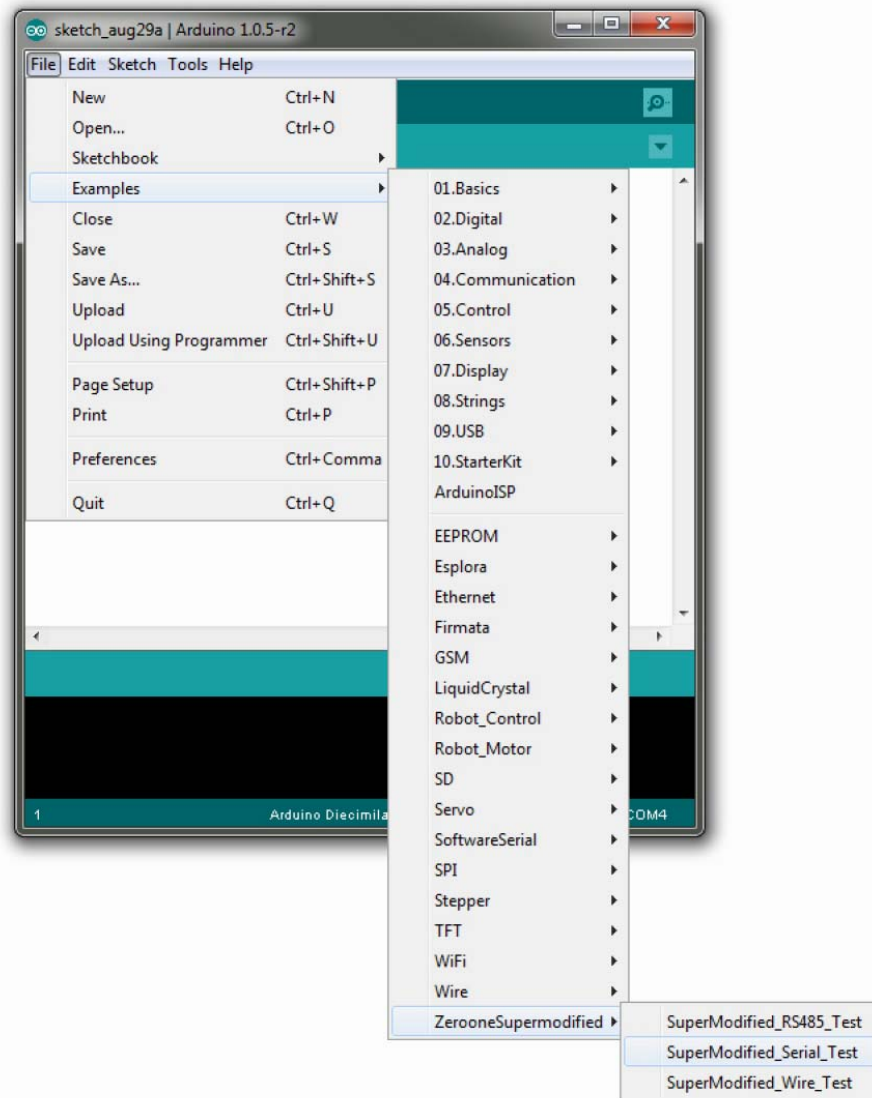
Compile and download the example. Then open the serial monitor and configure it at 57600 baud.

Before trying out some of the available commands, go ahead and check that the controller is currently not applying any force on the motor. Try to move the shaft of the motor and notice that it rotates freely. Now type 1 on the serial monitor and press enter.
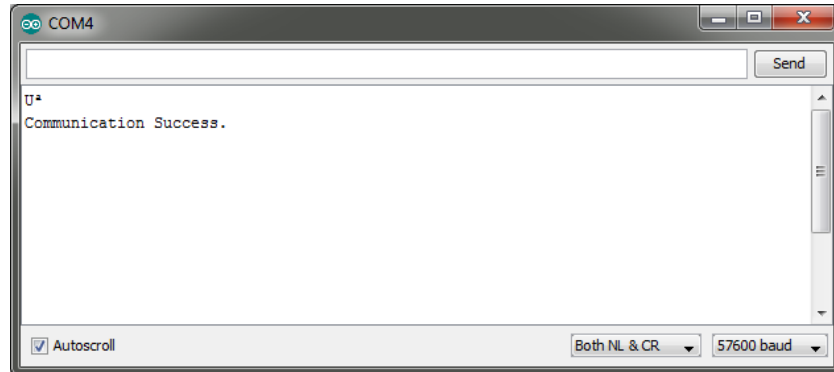
The motor now holds its position firmly. Type 3 and press enter. The motor moves with a velocity of 5000 ticks/sec. Type 4 and hit enter. The motor moves with a velocity of -5000 ticks/sec.

There are some strange characters on the serial monitor. They are the controller's response to the command just received. Because they are in binary and not ASCII format they do not display correctly on Arduino's serial monitor. You can monitor them on a terminal that displays bytes received from serial in hexadecimal representation.

The base unit for the 01^{TM}Supermodified controller is the encoder tick. The MagEnc utilizes an absolute encoder that has 32768 ticks per revolution. So in order to instruct the controller to do a full turn you must tell the controller to move by 32768 ticks. All commands utilize the encoder tick as a base unit. Thus velocities are in ticks/sec and accelerations in ticks/sec^2.

Testing with legacy RC-Interface: This is supported by the core Arduino library Servo.h

5. Warnings and errors

Try pressing 2 followed by enter and then 3 followed by enter. The motor does not move at all and at the serial monitor this message appears:
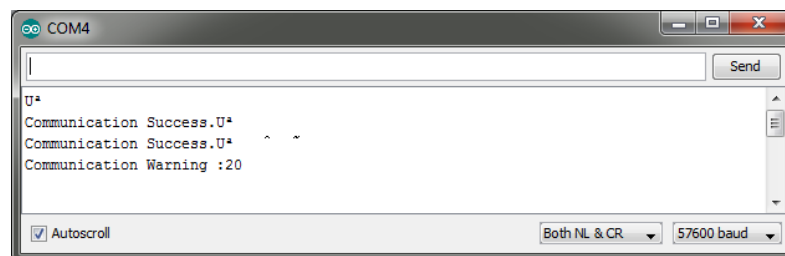
Communication warning 20 -> 0x14 in hex format. You can see what exactly this means in the error code reference section of the Supermodified datasheet:
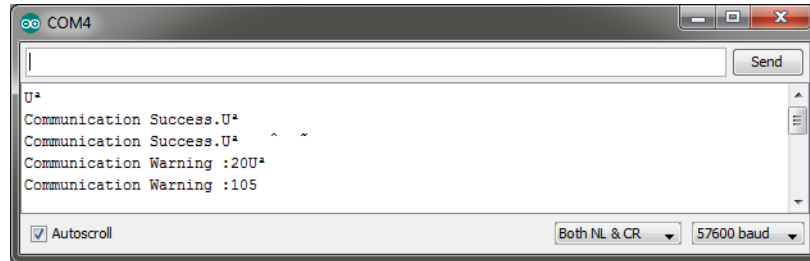
| 0x14 | Invalid command for motor state | DO NOTHING | *Description:* The received command is invalid for the given motor state. ie the 01^{TM} SuperModified controller is instructed to move the motor with a specific velocity prior to receiving a Start command (initialization and PID activation). The command is not executed. An error response is issued immediately. *Resolution:* Issue a Start command before attempting to issue movement commands. |

If you try to execute any command after that, it will fail with the same error code. This is

because the Supermodified controller needs to be explicitly instructed to erase the error. There are many good reasons for this and some of them are related to safety.
So if you want to continue using the controller you must issue a RESET ERRORS command.

Hit 5 and enter and then 1 and enter and 3 and enter. You have now reset the error, started the controller and instructed the motor to move at 5000 ticks/sec.

Now disconnect the Rx and Tx wires and try hitting 2 and then enter.

```
COM4                                                    [ - ] [ □ ] [ X ]
|                                                              [ Send ]
Uª
Communication Success.Uª
Communication Success.Uª      ^   ~
Communication Warning :20Uª
Communication Warning :105

☑ Autoscroll                          Both NL & CR  ▼   57600 baud ▼
```

The Arduino responds with Communication warning 105. Warnings with a code of 100+ are issued by Arduino itself. You can see what this means by referring to the library header file:
C:\Program Files\Arduino\libraries\ZerooneSupermodified\ZerooneSupermodified.h
or https://zerooneltd.freshdesk.com/solution/articles/5000653187-arduino-library-error-list

```
/*
    ZerooneSupermodified.h - ZeroOne Supermodified Controller API
    for Arduino.
    Copyright (c) 2014 ZeroOne Mechatronics.  All right reserved.

    This library is free software; you can redistribute it and/or
    modify it under the terms of the GNU Lesser General Public
    License as published by the Free Software Foundation; either
    version 2.1 of the License, or (at your option) any later version.

    This library is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
    Lesser General Public License for more details.

    You should have received a copy of the GNU Lesser General Public
    License along with this library; if not, write to the Free Software
    Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA  02110-1301  USA
*/

#ifndef ZEROONE_SUPERMODIFIED
#define ZEROONE_SUPERMODIFIED

#include <Arduino.h>
#include "utility/zoProtocol.h"

//Configuration defines
#define ZO_PROTOCOL_COMMAND_RESPONSE_TIMEOUT_MS          200

//--Warnings-----------------------------------------------------------
#define ZO_WARNING_NONE                          0
#define ZO_WARNING_WIRE_PACKET_OVERWRITTEN       100
#define ZO_WARNING_WIRE_BUFFER_NOT_ENOUGH        101
#define ZO_WARNING_WIRE_NO_ACKNOIWLEDGE          102
#define ZO_WARNING_WIRE_BUS_ERROR                103
#define ZO_WARNING_WRONG_LRC                     104
#define ZO_WARNING_RESPONSE_TIMEOUT              105
```

In this case a response timeout has occurred. This means that the Supermodified did not issue a valid response within the predefined time window of 200ms (…RESPONSE_TIMEOUT_MS).

6. Arduino as a USB <> UART / RS485 bridge

If you have an Arduino that has an IC socket and the microcontroller can be removed it is possible to use the board without the microcontroller as a USB<>UART or USB<>RS485 bridge. For UART or RS485 the connections are illustrated below:
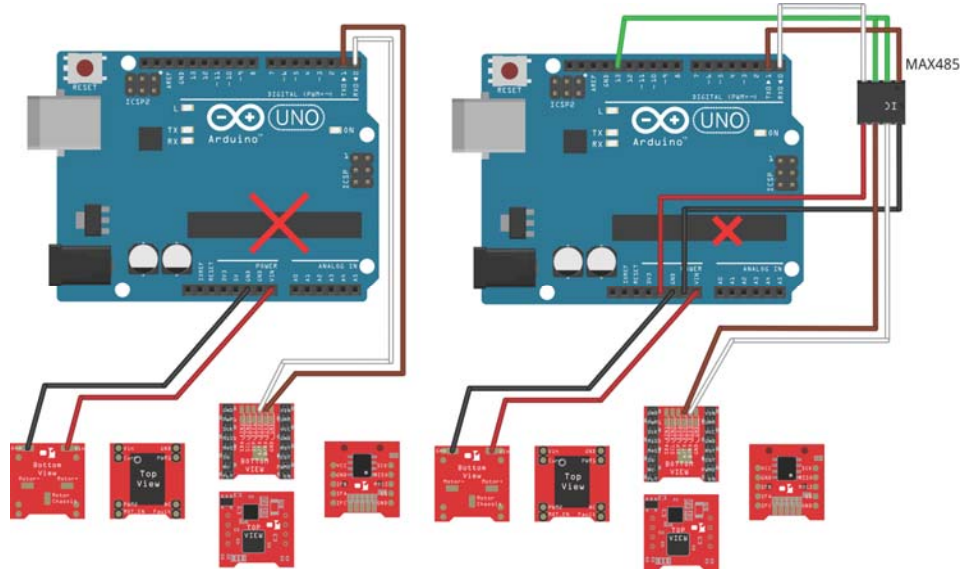
With these connections you can use your Arduino board exactly as an USB<>UART / USB/RS485 bridge. Please refer to Testing with USB<>UART on how to easily send commands to the Supermodified controller.

7. Contents