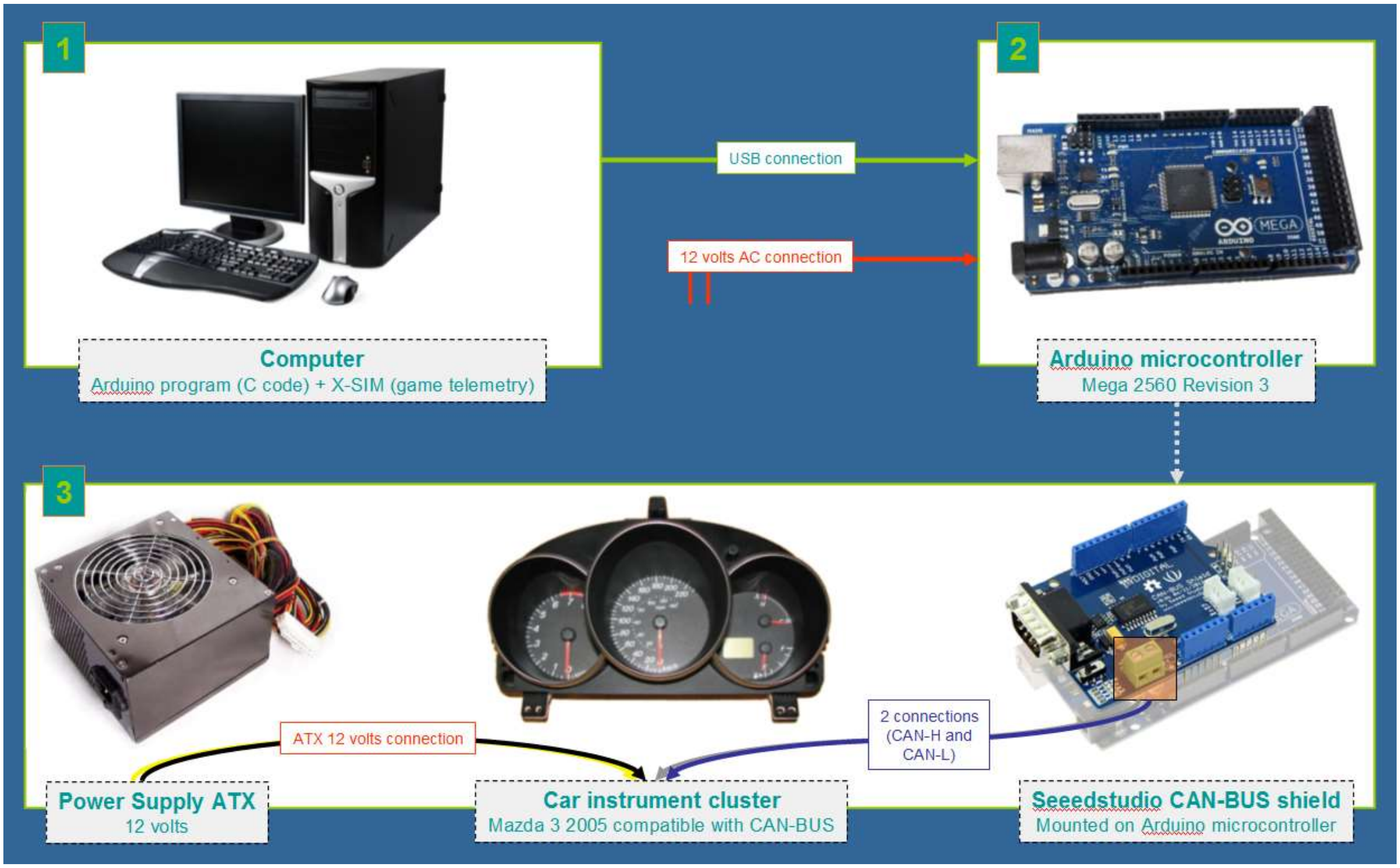
Build a real car instrument cluster for racing simulation on a PC

Components required:

1- PC computer with Windows 7, Windows 8 or Windows 10
2- Arduino Mega 2650 revision 3 micro-controller
3- Seeedstudio CAN-BUS shield
4- 12 volts ATX Power Supply (equipped ideally with a an interrupter)
5- AC/DC power adapter with 1-2 amp 12 volts output
6- Mazda 3 2005 instrument cluster including its two cable harnesses
7- Program like Sim-Tools or X-Sim to feed the micro-controller
8- Arduino program to upload the micro-controller with the proper code

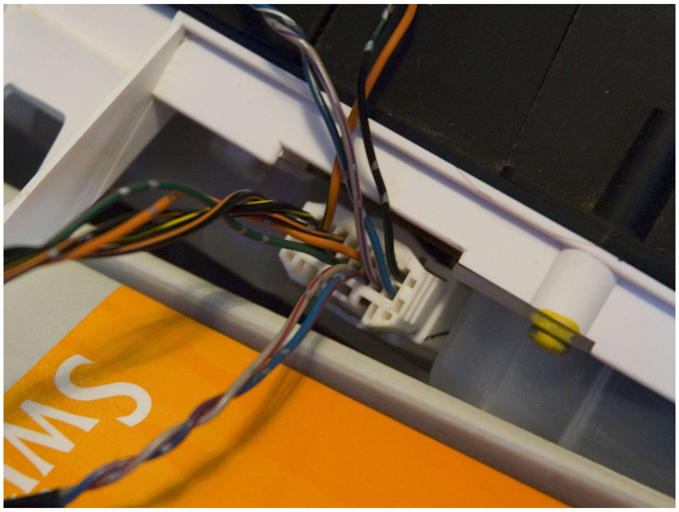
General connection diagram:


1
USB connection
2
12 volts AC connection
Computer
Arduino program (C code) + X-SIM (game telemetry)
Arduino microcontroller
Mega 2560 Revision 3
3
ATX 12 volts connection
2 connections (CAN-H and CAN-L)
Power Supply ATX
12 volts
Car instrument cluster
Mazda 3 2005 compatible with CAN-BUS
Seeedstudio CAN-BUS shield
Mounted on Arduino microcontroller

Instructions:

The following instructions will guide you through the many steps involved to properly connect the components mentioned earlier. The included pictures will help you to get the right configuration. With the proper game telemetry plugins the cluster should react to all the inputs related to speed, rpm and gear shift.

Step 1



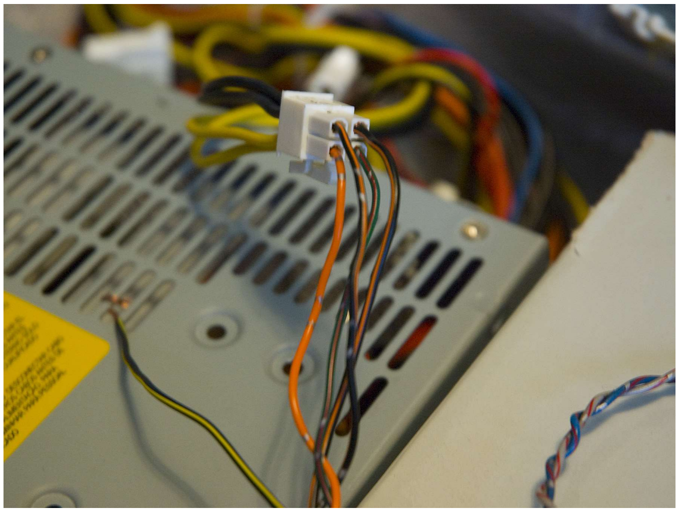
Pick the smaller cable harness, seat it in the proper port of the instrument cluster and isolate the four following wires:

Black/Orange, Black/Orange, Green/Red, Orange, Black/Yellow

Isolate the two Blue/Red and Grey/Red wires and twist them to form a twisted-pair cable on the whole length of the wire.

Remove 2 cm of sheath from the end of each seven cables. Fold in half the bare part of each cable except for the Black/Yellow one.

Step 2

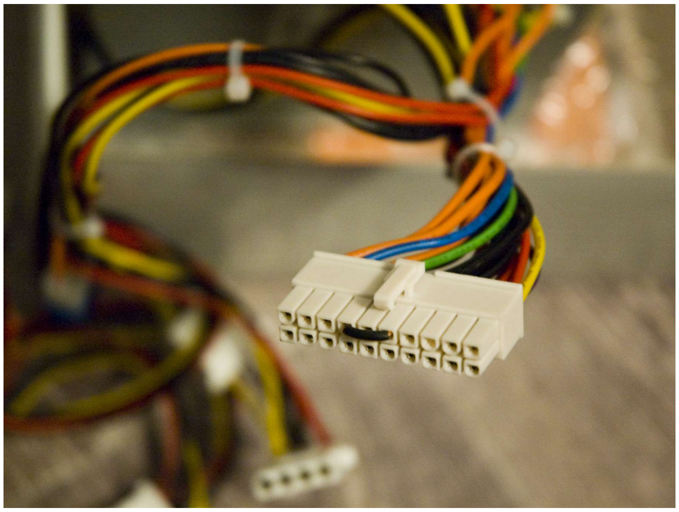


Take the 12 volt connector of the power supply and identify the two black-wired ports and the two yellow-wired ports.

Insert the folded bare ends of the two Black/Orange cables in the two black-wired ports, and both Green/Red and Orange cables in the two yellow-wired ports. Add electric tape around the joined wires to secure the connection.

Attach the bare end of the Black/Yellow cable to a metal part of the power supply for ground.

Step 3

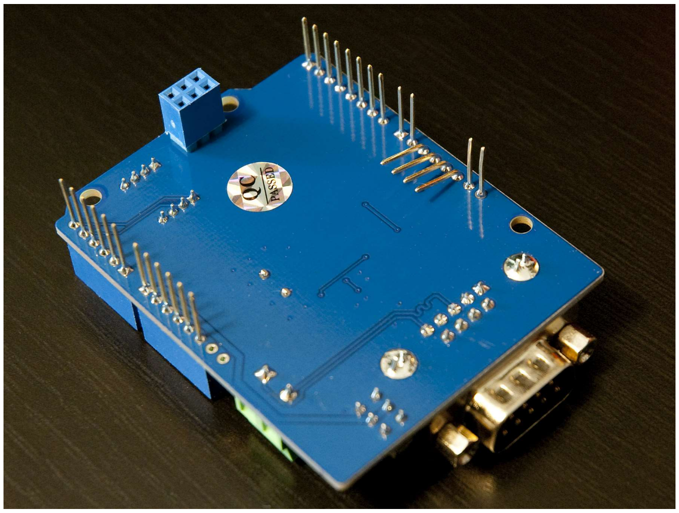


Take the 20-pin connector of the power supply and jump-connect the green and black-wired ports in order to start the power supply as soon as it's connected to a AC wall plug.

If the power supply is equipped with an interrupter that will allow to leave the cluster and the power supply connected to the AC outlet without always being in function.

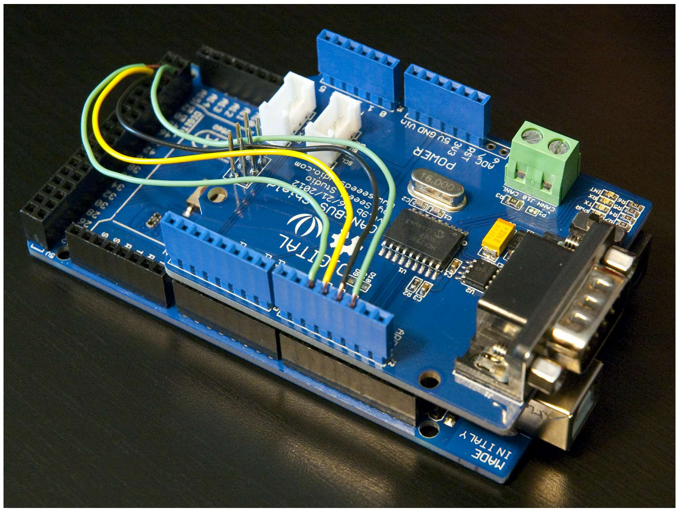
Cover the 20-pin connector with electric tape.

Step 4



Take the CAN-BUS shield and place it upside-down. Locate the 4 pins numbered 10-11-12-13 and bend them delicately with tweezers like depicted in the picture above.

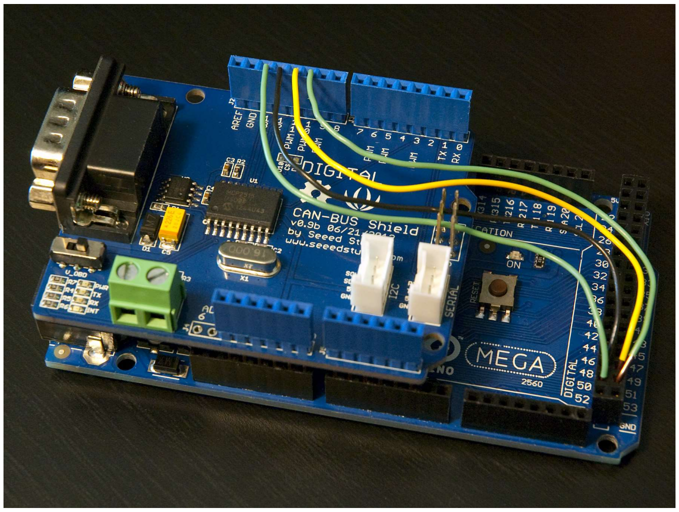
Step 5



Install the CAN-Bus shield on top of the Arduino Mega board.

Prepare four wires of 10 cm each and remove 1 cm of sheath at the tip of each cable. Bend in half the bare ends of the wires.

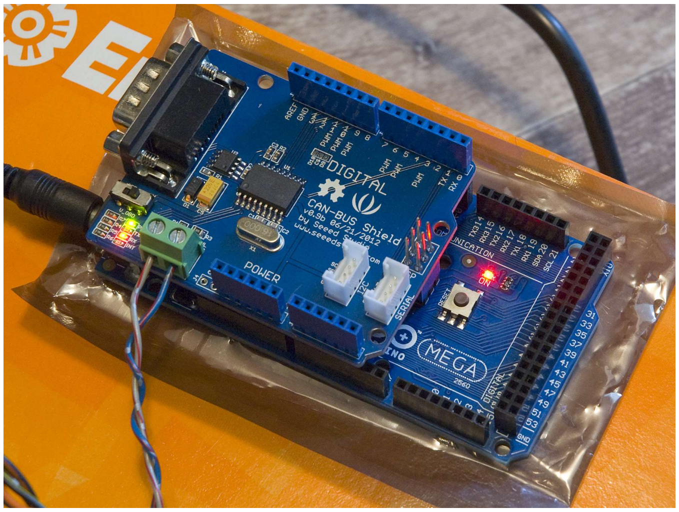
Step 6



Create a special bridge between the CAN-BUS shield and the Arduino by connecting the 4 wires in the following sequence:

CAN-BUS port 10 to Arduino port 51
CAN-BUS port 11 to Arduino port 49
CAN-BUS port 12 to Arduino port 48
CAN-BUS port 13 to Arduino port 50

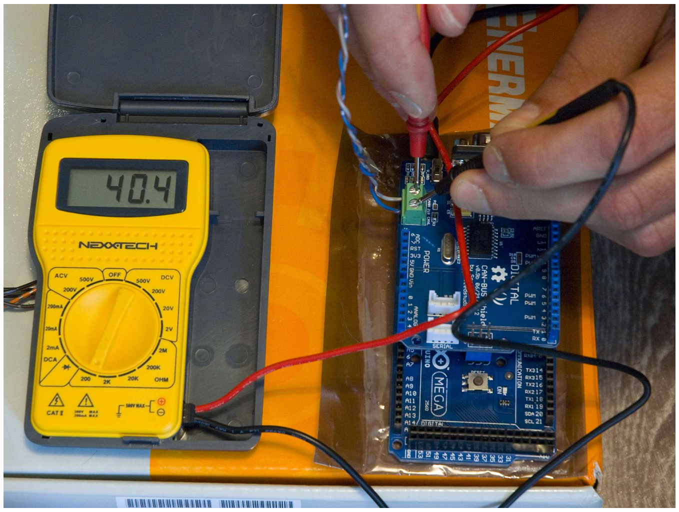
Step 7



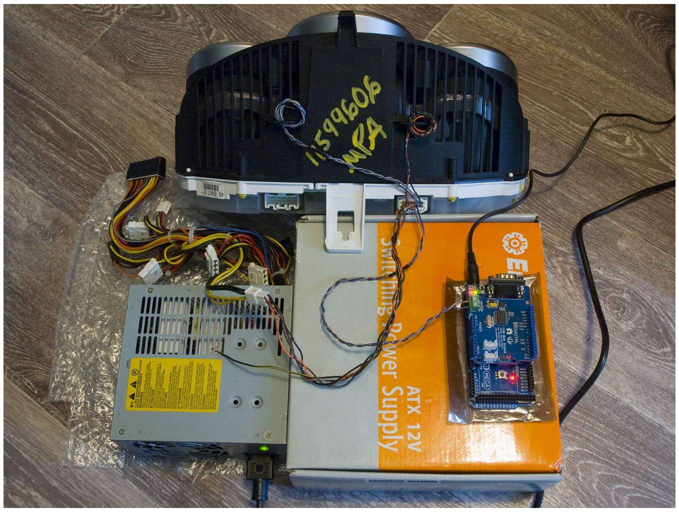
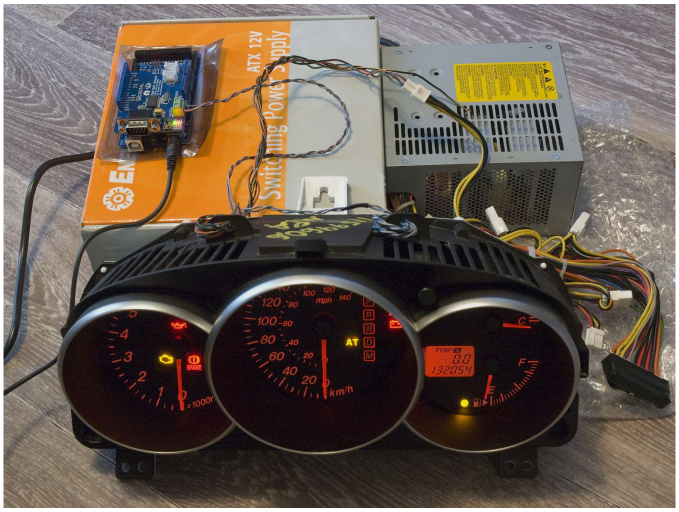
Take the twisted-pair cable and insert the bare end of the Grey/Red wire in the CANH port of the CAN-BUS shield. Insert the bare end of the Blue/Red wire in the CANL port. Secure the wires with the port screws.

Note: The picture above was taken before the installation of the four wires providing the CAN-BUS/Arduino bridge and it is not showing the final configuration. Same goes for the next pictures. Keep in mind that at this stage the four wires should be there.

Step 8



The resistance on the CAN-BUS shield CANH and CANL ports should indicate 40 Omhs.

Step 9	**Step 10**
 The final configuration should look like in the picture above once all the wires and components are connected. At this stage you can connect the Arduino board to a USB port of the PC and to a wall outlet using the AC/DC 12 volt power adapter.	 Now turn on the power supply and verify that the cluster is doing is starting routine. The needles should go all the way up and then down.

The next steps consist in configuring the Arduino micro-controller to make it work with the pc platform:

Step 1:

Download the latest Arduino driver for the COM3 port (Windows 10 should already have the driver pre-installed):
http://www.driverscape.com/download/arduino-mega-2560-r3-%28com3%29

Visit the Arduino website for more info about initial configuration and to download the software package:
https://www.arduino.cc/en/Guide/ArduinoMega2560
https://www.arduino.cc/en/Main/Software

Step 2:

Download the library for the CAN-BUS shield:
http://wiki.seeed.cc/CAN-BUS_Shield_V1.2/

Step 3:

Follow all the instructions in the above websites to properly configure the Arduino/CAN-BUS boards and install the software/libraries onto your PC.

Step 4:

The Arduino sketch to program the micro-controller was initially developed to work with X-Sim environment but a version of the codes compatible with SimTools will be under development and available soon. First I need to get a DIY licence of SimTools in order to develop the codes that will work with most racing games. If anyone of the community is already having a DIY licence and wants to give a shot at developing the corresponding SimTools code based on the original X-Sim code, you are welcomed to visit the following links to get access to the Arduino sketch and the X-Sim configuration.

OneDrive link to Instructions in PDF format, hi-res photos, Arduino sketch and X-SIM configuration:
https://1drv.ms/f/s!AtwKGzBk3VOohJdX5ZtdmwFRk_GPcw
Note that the Arduino code contains extra programming lines for a shift light and a 7-segment LED display showing gear positions and digital speed