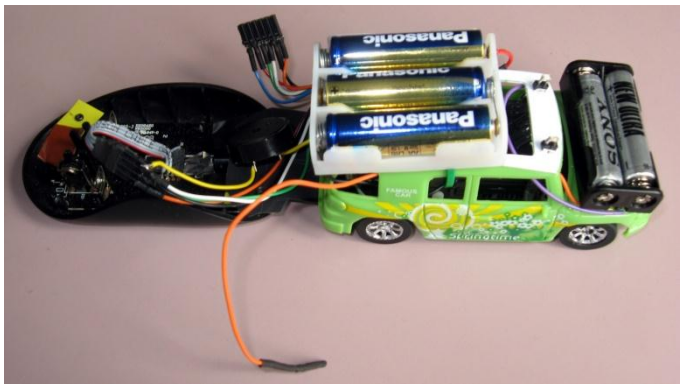
User Manual

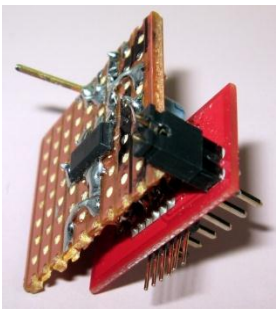
Provided Elements

The refitted toy car set includes:

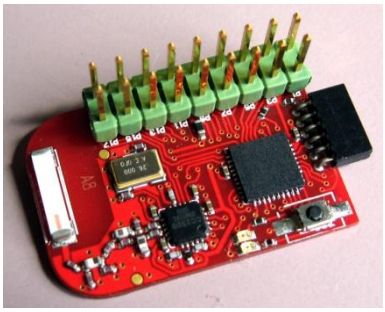
1. The refitted car with the mouse and battery holder attached



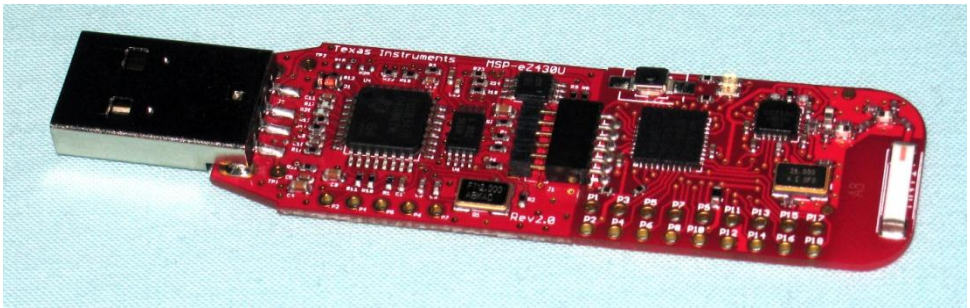
2. The power regulation circuit



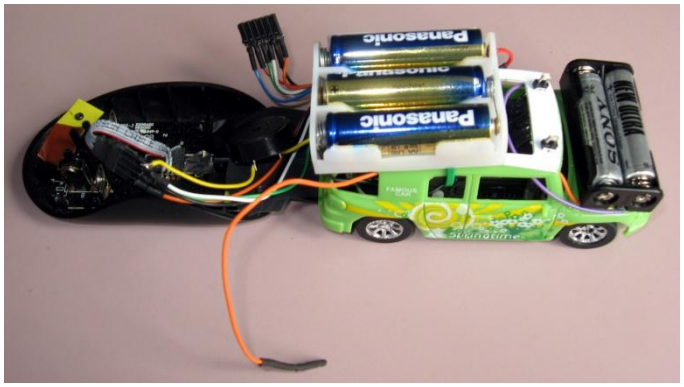
3. Car Serial Wireless



4. PC Serial Wireless



Toy car

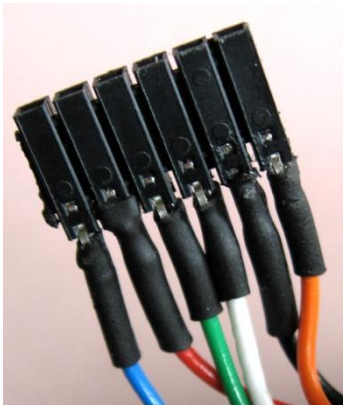


Battery

The toy car has two sets of batteries. The top holder contains 3 AA batteries that power the car original circuit. Those 3 batteries supply the car motor and steering coils. The second set powers the embedded system. The set is composed of the front battery holder and the bottom battery holder from the car. Two AAA batteries are placed in the front holder and **1 AAA is placed in the bottom holder**. The battery in the bottom holder is placed in the slot way from the holder screw.

WARNING: Care should be taken when placing the batteries into the holders.

Connections



The car has two loose connectors. The connector shown here is the embedded system connector and connects to the *Power Regulation Circuit*. The following signals are present:

- UART Tx : Blue
- VCC (3.3V) : Red
- TST : Green
- RST : White
- GND : Black
- UART Rx : Orange

WARNING: Care should be taken when connecting this connector

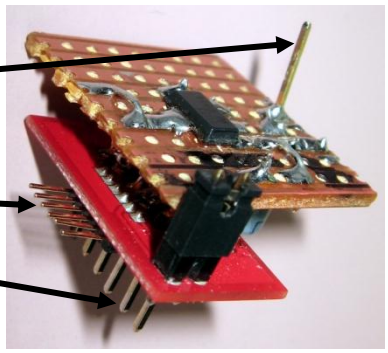
The second connector is the loose black-ended orange wire. This wire is connection to the embedded system battery set. It provides a nominal voltage of 4.5V to the *Power Regulation Circuit.*

Power Regulation Circuit

The power regulation circuit generates a stable voltage for the embedded system from the embedded system battery set. The black-ended orange wire from the car connects to the single pin on the top.

The power regulation circuit interface the *Car Serial Wireless* with the embedded system. The *Car Serial Wireless* connects to

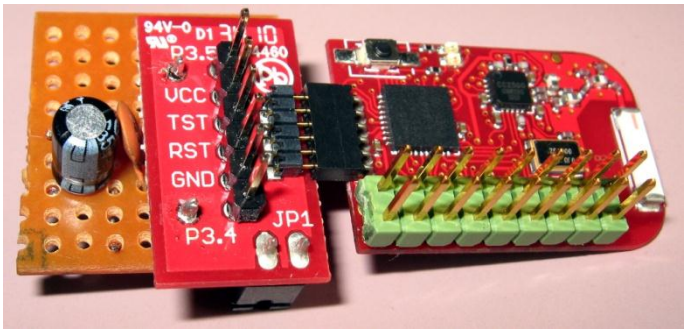
The circuit connects to the car embedded system connector with Labels are present to correctly connect the circuit.



WARNING: Care should be taken when connecting the circuit to the car connector.
Failure to connect the circuit correctly may result in the embedded system being damaged.

Serial Wireless

The toy car communicates with the computer via a serial link. The serial settings are 9600 baud, 8 bit data, 1 bit stop, no parity bit.

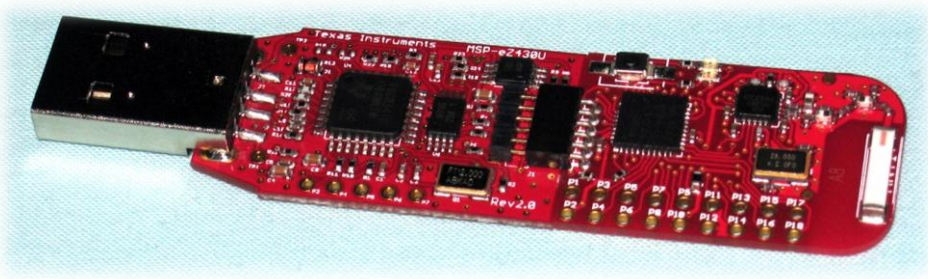


The serial link can be achieved wirelessly by Texas Instrument RF2500 which simulates a serial link proxy.

The *car serial wireless* connects to the *power regulator circuit* as shown above. The *car serial wireless* circuit enables the serial link once the switch is pressed. Then the *PC serial wireless* is able to connect and exchange serial commands.

The *PC Serial Wireless* is shown below. When plugged, the dongle waits for the connection to be enabled by the car circuit. This state is indicated by 2 blinking LED. Once the connection is established the LEDs indicate transmission and reception of data.

If the cars wireless circuit is reset. The connection must be reinitiated by reconnecting the dongle to the computer and enabling the link for the car circuit.



Software Setup and Install

The software for this project was developed using the Code Composer Studio (CCS) IDE and is freely available to download [1]. It can be used to develop and port code to many Texas Instrument microcontrollers including the MSP430.

All the code, along with version history, can be found and downloaded from the EE579 project Google repository [2].

Once CCS V5.3 and the Project code have been downloaded you are ready to begin the installation process.

1. Open CCS and Navigate to File > Import
2. In the newly opened dialog window select Code Composer Studio > Existing CSS Eclipse Projects and then press Next.
3. Select Browse and navigate to the ‘ToyCar’ folder located in MicroController > EE579WS folder found in the project folder downloaded from the Google repository.
4. Select Finish

Now that the code is imported into CSS it can be transferred to the MSP430 located in the car. To do this connect the Launch Pad, as shown in the Hardware Section/User Manual, and connect a USB from the computer to the Launch Pad.

Before porting the code to the MSP430 a few properties have to be checked and ensured to be correct otherwise the code will not compile. Right click the ToyCar folder in the CCS workspace and select properties.

In the General options,

- In the Main tab ensure that the *variant* selected is 'MSP430G2553' and the *compiler* version chosen is 'TI v4.1.2' as shown in Figure 1.
- In the RTSC tab ensure that 'Grace 2.0.1.65', 'MSP430ware 1.25.0.30' and 'Other Repositories ${TARGET_CONTENT_BASE}' are selected as shown in Figure 2.

Once the properties have been checked right click on the ToyCar folder once again and select Debug As > Code Composer Debug Session and the code will be compiled and ported to the RC car.

To use the car for its designed purpose the Launch Pad should be disconnected and the wireless serial connections should be connected to the car and the computer, see in Hardware section above.

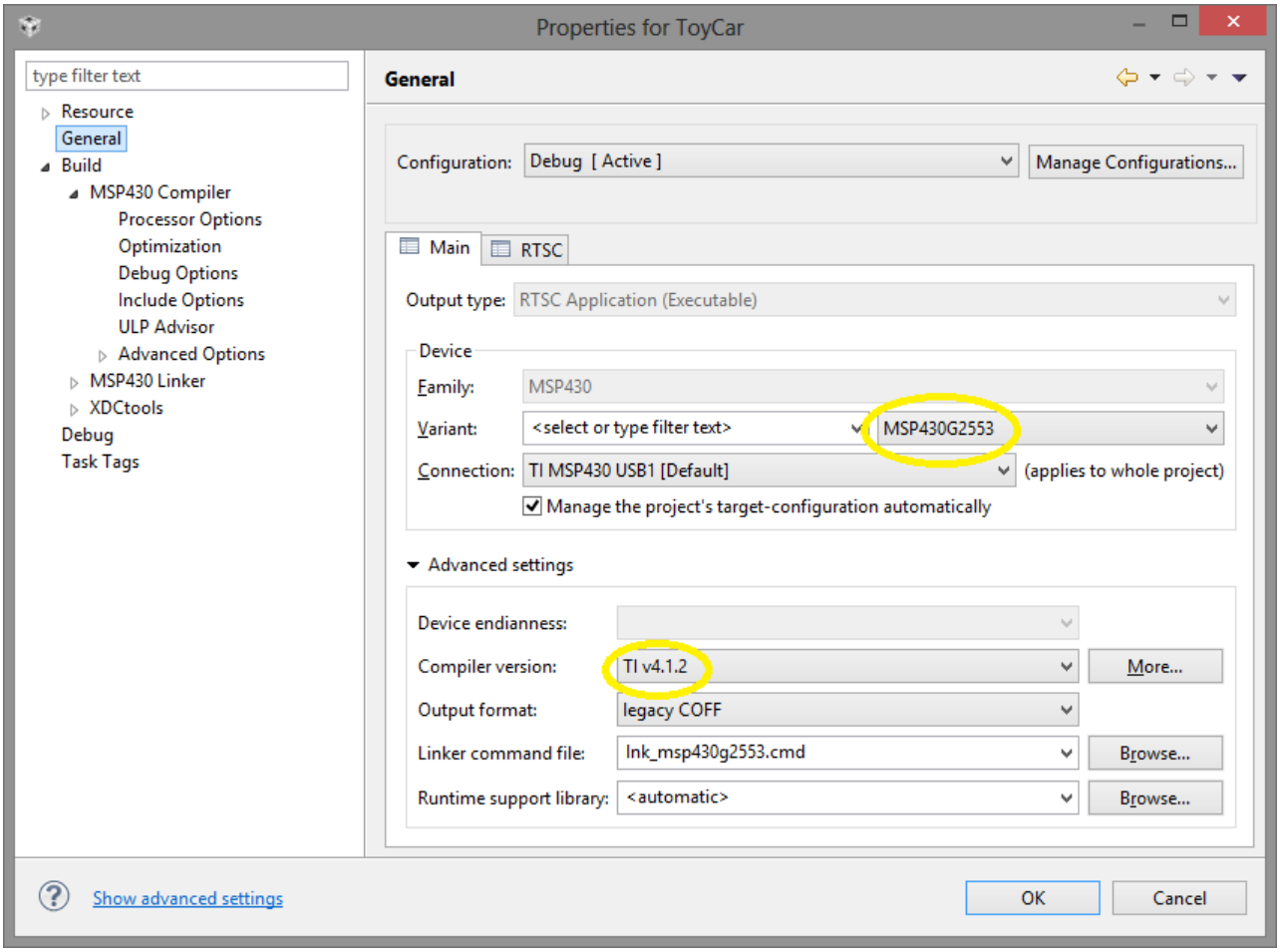


Figure 1 - The microcontroller variant and compiler version.

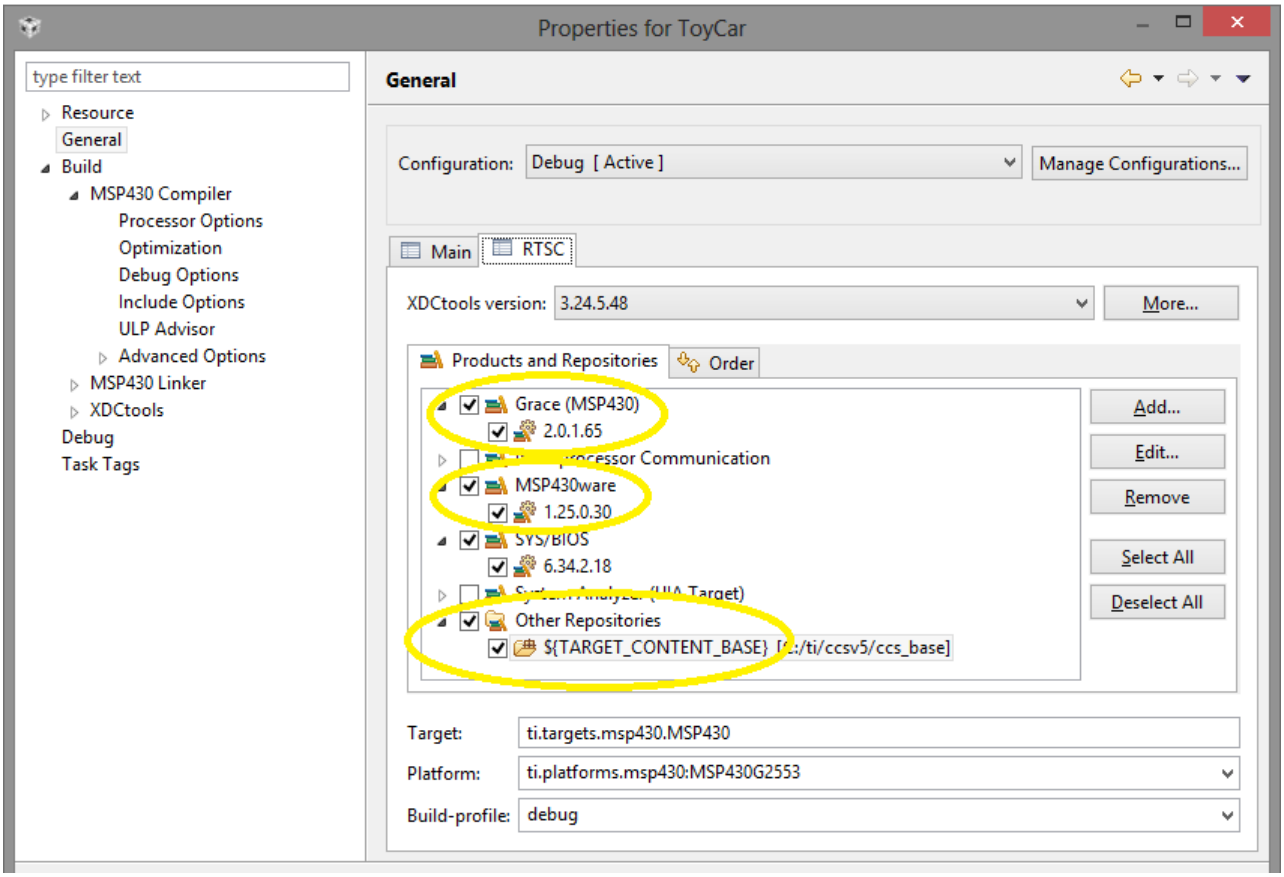


Figure 2 - Grace, repository and MSP430ware settings.

Software user manual

At this point, the car should have the wireless serial connected and the lights on its roof flashing, indicating that the microcontroller is running. The next step is to connect one's computer to the car through serial. In order to do so, the USB Dongle has to be plugged in and a channel has to be opened between the computer and the MSP430 using serial communication software like PuTTY. The BAUD rate to be used is 9600 without parity and the COM port to use is usually COM9. If it is not the right COM port, the port can be checked in Device Manager (Windows) the port number.

The serial channel establishment can be checked by pressing 'Space'. If the channel is active, the message 'Stop' should be displayed in the serial console.

To make the car move, the cars speed can be entered by pressing 'c' and entering an integer number related to the speed (15 is the recommended working speed) and then pressing 'Enter'. To enter the distance and angle travelled by the car, 'p' should be pressed. Firstly, an integer value representing the number of centimetres for the car to travel before turning is entered (and then press 'enter') followed by the angle (in degrees) that the car should turn through at the turning point. It is also possible to play music depending on the state of the car (before the turn, after the turn or at the finish point) by pressing 'm' before the 'p' stage.

The car will stop by itself after 30 seconds and it can be stopped manually anytime using ‘Space’.

References

[1] Texas Instruments, “Download Code Composer Studio,” [Online]. Available: http://processors.wiki.ti.com/index.php/Download_CCS.

[2] Google Code Project Page, “Adv Micro Project EE579 2013,” [Online]. Available: https://code.google.com/p/adv-micro-project-ee579-2013/. [Accessed 2013].