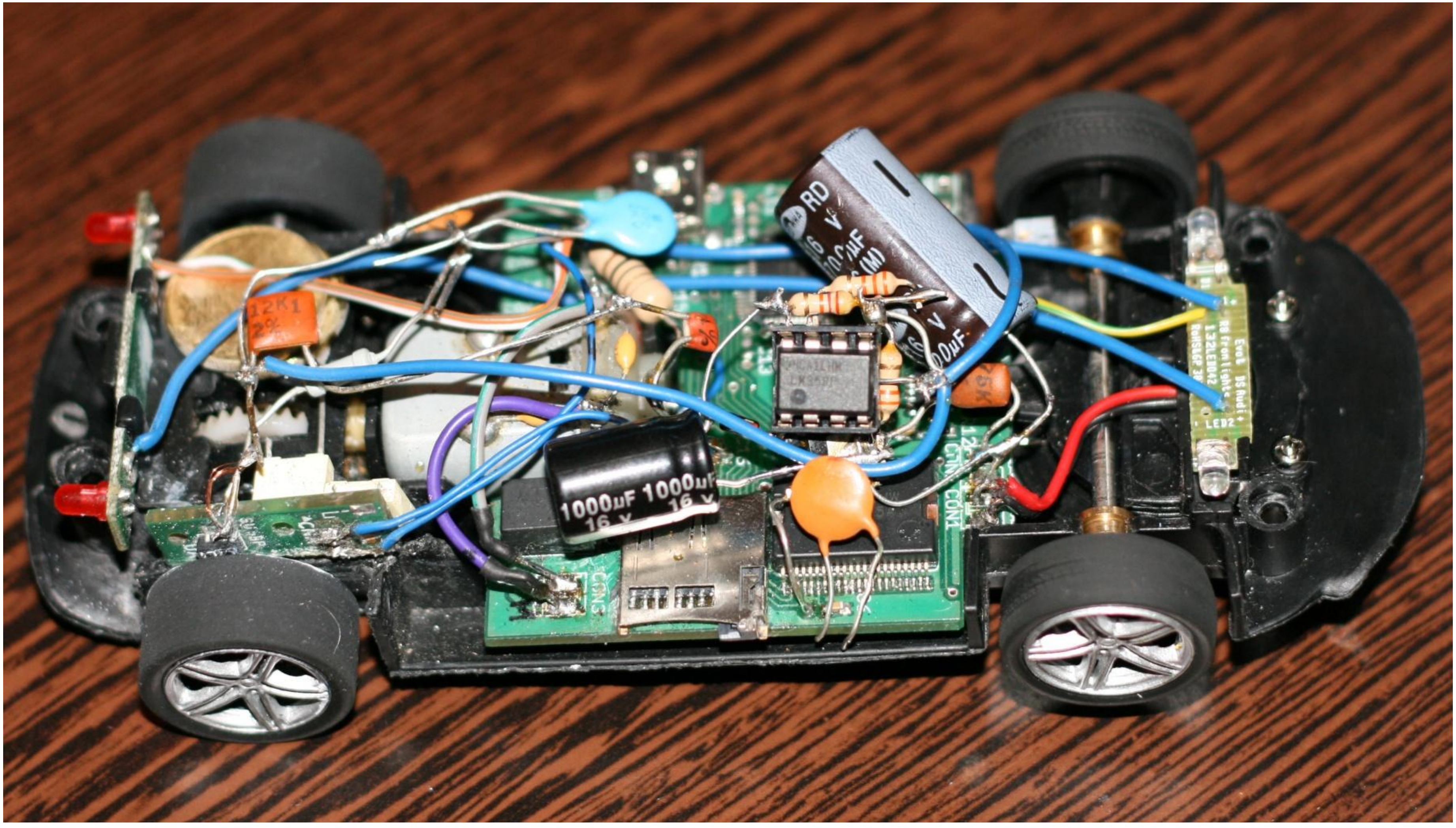
# E40 -Roadkill



# Custom Changes:

- Changed the low pass filter for sensing acceleration on X axis
- Added an optic sensor for distance and speed measurement
- Added a Schmitt trigger for setting custom values for the sensor’s output signal
- Grinded the back tires for minimum skid and drift
- Grinded the front tires for smoother driving
- Added extra weight up the 125g limit

# Interrupts used:

•TPM1 Channel 0, Negative Edge Detection for Distance and Speed measurement
•RTC, 125us for algorithm decisions
•TPM2 Overflow, for motor PWM generation
•ADC, 1/2ms, for accX and trackVoltage sampling

# microSD Card Operations:

•We found that the FAT functions were very time and resource consuming so we used the raw disk i/o system instead
•A 512 Bytes buffer is used for temporary data storage
•We initialize the microSD card using the Disk_Initialize() function
•Writing the buffer to a raw sector on the card is done using the Disk_Write() function
•Reading a raw sector from the card into the buffer is done using the Disk_Read() function

•This is a lot faster (and safer, considering the difficult mechanical conditions in which we use the card: lots of power losses, vibrations, acceleration shocks etc)

# I. Optic Sensor

We added an optic sensor that monitors the rear right wheel. The wheel is split in 8 complementary areas. Depending on which area is in front of the sensor, it’s output is either logical 1 or logical 0. Each time one white area passes, the car has traveled 1.65 cm. At each pass an internal timer is also reset. Using that timer the speed is computed.

# II . Acceleration filter

Our algorithm uses a custom IIR filter for monitoring accX. We developed a second order Butterworth filter. This way the acceleration's samples have a delay of just 40 ms. The filter is described by the following equation:

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

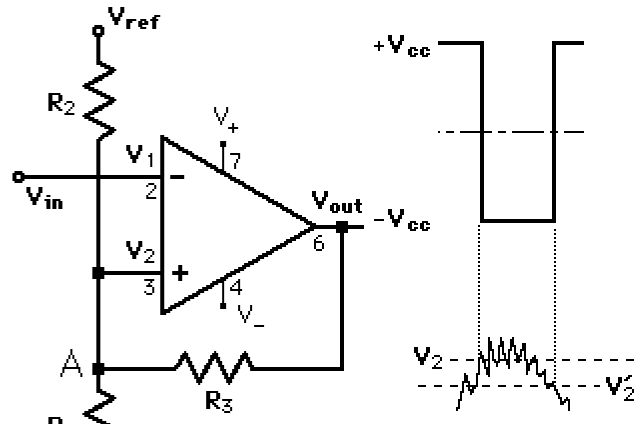
$$y(n) = b_0 x(n) + b_1 x(n-1) + b_2 x(n-2) - a_1 y(n-1) - a_2 y(n-2)$$

# III. Schmitt Trigger

We added a custom made Schmitt trigger circuit. The circuit was made out of an UA7805, a 1N4148 diode, a LM358 operational amplifier and resistors. This way we were able to set custom voltage levels for the logical 1 an 0 outputted by the optical sensor, creating hysterezis:

$V_2 = 2.5$ V

$V'_2 = 1.5$ V




$$v_2 = \frac{R_{123}}{R_2} V_{ref} + \frac{R_{123}}{R_3} V_{cc}$$

$$v'_2 = \frac{R_{123}}{R_2} V_{ref} - \frac{R_{123}}{R_3} V_{cc}$$

# IV. Algorithm

## A. Learn

The learn algorithm maps the track starting with the first lane change it encounters. The lane change is detected when certain conditions are met. The most important is that the track voltage has to drop near 0 V. On top of this the algorithm checks for other details that can rule out false lane changes which can occur when the track isn’t assembled correctly.

After the first lane change the algorithm looks for threshold breaching on the X acceleration and maps the track accordingly.

The mapping is finished when the third lane change is detected. At this point the map is written in a raw sector on the SD card.

The map is an image of the track split in curves and straight lines. For each of these, the map contains the portion type (straight line, left curve, right curve), the starting point in the track, and it’s size. On top of this information the map also contains the position of the lane changes and their type for future synchronization.

# B. Drive

At this point the algorithm has a map of the track. When Drive mode is enabled the algorithm sets a certain maximum speed according to the next track portion.

For the long straight lines (longer than 40cm), the speed is ramped up to the maximum value. The starting point for the acceleration is computed based on the distance measurement.

For the short straight lines and short curves, a constant motor voltage is maintained, up to a safe value of the speed.

For the long curves, the speed is also ramped up but it’s maximum value is set lower than in the case of long straight lines. The ramping is also slower to minimize skid.

# C. Synchronization

In Drive mode a sync mechanism is enabled. At every curve and straight line the distance parameters are synchronized with the map available. This means that the distance measurement are correlated with the acceleration measurement. This way the skid is neutralized.

# Team Roadkill:

Andrei Gheorghe, ghiga_andrei@yahoo.com
Andrei Scurtu, andrei_scurtu@ymail.com

Faculty of Electronics, Telecommunications and Information Technology
Polytechnic University of Bucharest, Romania

30.04.2010