AROS REFERENCE SHEET
THE FREESCALE CUP 2014

Overview
- Operating voltage: 6-7,2V
- Max current: 7,7A (blocked motors)
- CPU: Freescale K40x256
- Sensors:
  - 2 line scan cameras
  - 2 tachometers (only one used)

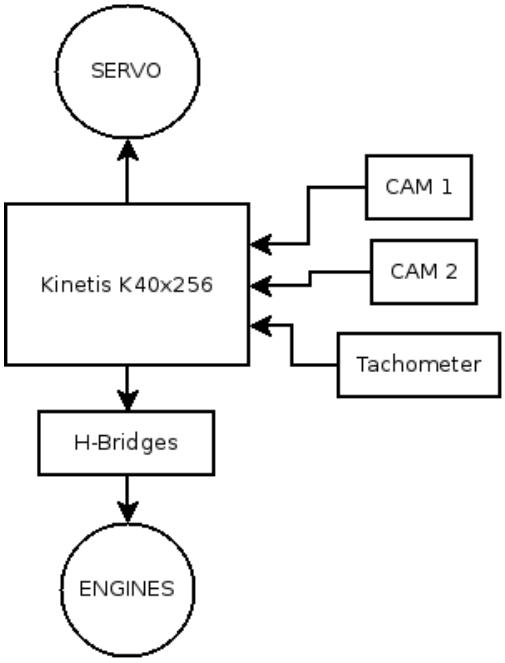
Hardware
Was used standart TFC kit with following modifications:

a) Chasis

On chasis was mounted platform for boards and construction for cameras

b) PCB

Besides TWR kit with K40, drives board v.1 with reverse upgrade was added boards with tachometers and board with aditional stabilisers with total capacity 40uF



Software
a) Structure

Software structure of AROS is made from three main parts. It's main loop, support functions (like setup of FlexTimers, ADC etc.) and camera processing algorithms.

b) Main loop

On top of main before loop begins is located init part. It it are called support functions. They are described at part c). Main loop is infinite loop where are all calling for drive algorithms. Calling of them is initiated by finishing of read from camera array.

c) Support functions

Support functions (directory called CPU) contains following:
- Port setup
- FlexTimer setup with interrupt implementation
- Clock setup
- ADC with interrupt implementation
- Cortex M4 dependencies

d) Camera processing

Function Findline

Findline function is constructed to detect black line on white background. Camera stores values of scanned voltage into integer array, then this function counts average value of the camera output and compares each value with this average treshold. If it detects descending edge in the signal (left value is above the treshold and right value is below it), it stores position of this treshold and continues. If it finds raising edge, then the position of it is stored again and by average between these two points actual center of the line is counted and returned as returning value from the function.

Function Zpracuj

Zpracuj function will scan few last detected values from the camera and compare it with ideal center position. If these values are not begind the tolerance tresholds and are most likely the same (with a little tolerance), the car won't slow down. If steered wheels are in center position, car will accelerate. Car will slow down, if last few line positions from camera are outside of defined boundaries or if subtraction of highest and lowest value is too high, which would mean the line is getting lost.

e) Implementation

FlexTimer generates clock for camera at frequency 70kHz. ADC reading routine responsible for reading data from camera is called on every second timer's tick (only on rising edge). After AD conversion is finished, it's interrupt routine is called and read value is stored into camera buffer array. Also is incremented pixel position up to 128.

In pseudo-parallel run with this is main loop, where are calls for ADC (as described before) and also is there condition after reading is finished (pixel counter is set to 128). It launch image processing and then decision algorithms. After all that is set pixel counter to 0 (which enable writing into camera buffer) and is launched Start impulse (SI) for camera to begin next round.

Implementation of PWM for servo and engines is made from one timer and few conditions, so we are not limited by hardware specified PWM channels.

Team

- Vojtech Vladyka
- Jan Zlebek
- Jiri Stavelik

supervisor: Ing. Miloslav Richter, Ph.D.