Up to date 26/November/2024; 8pm; signed by Sam.

NOTE:

1. Vsetky funkcie, ktore mame su v .h hlavickovych suboroch.
2. Dole je vypisane ako ku datumu kod postupuje a co sa vola. Bolo by super z toho spravit vyvojovy diagram ☺. Neslubujem, ze si nato najdem cas..

# File structure:

Include + Source:

1. BASIC
   1. ADC.h/.c
      1. ADC operation, filters for ADC, interrupts for ADC
   2. LED.h/.c
      1. LED control, start-up, braking
   3. Motor.h/.c
      1. Motor control, braking, reversing, PWM control
   4. UART.h/.c
      1. Bluetooth over UART control
      2. Cyclic buffers for sending char, uint and int
      3. Command line with basic commands
   5. main.h/.c
      1. Initialisations of peripherals and functions
      2. Main while loop
      3. Command line control
2. ENHANCED
   1. Timers.h/.c
      1. Time-out flags
      2. Time-out variables
      3. Init main clock
      4. Init timer A0, A1, B0
      5. Variable delays
      6. PWM signal generation
      7. Interrupts for time-out features
   2. Flash\_memory.h/.c
      1. Manage flash memory writes/reads
   3. Data\_temp\_storage.h
      1. Storage of constant values for off-board testing
   4. Data\_operation.h/.c
      1. Correlation algorithms
         1. Auto-correlation
         2. Exact Match
         3. Sum of Absolute Differences
      2. Correlation windows handling
      3. Correlation dumping incorrect samples
      4. Correlation thresholds
      5. Correlation clearing buffers
      6. New-lap-finding functions
      7. Convolution functions
      8. Complex Bluetooth user debug handling
   5. Aux.h/.c
      1. Simple car control
         1. Different axis handling
         2. LED handling
         3. Speed handling
         4. Bluetooth debug for Phyton script
      2. FSM car control
         1. Handling Ref-lap, Run-lap, Stop, Error, Debug, Recovery, Reset states
         2. States transmission
         3. Z-axis handling
         4. LED handling
         5. Map saving
         6. Correlation performing
         7. Proper braking LEDs handling
         8. Complex speed handling
         9. Complex map creating
         10. Recovery and reset
         11. Complex debug possible
         12. Possible training on pre-saved data
         13. Lap counter
      3. MAP
         1. Map saving
         2. Map showing
         3. Map creating (in progress)
         4. Map clearing
      4. Complex Bluetooth user debug handling

# Start-up procedure

1. After start up -> main.c = init of all peripherals, infinite loop handling the program
2. Timers are handled in timer.h/.c
3. ADC readings are in ADC.h/.c
4. Bluetooth handlings are in UART.h/.c
5. -> Goes to aux.c = all functions for car controlling. If FSM init -> first REF\_LAP FSM state
6. If calling correlations -> jumps to data\_operation.c = all data operation functions
7. -> aux.c handles positive correlations and proceeds to the next FSM state
8. According to the time-outs and flags proceeds with ADC readings, LEDs controlling.
9. Saves map with every new ADC sample
10. After new lap found, the speed is controlled according to the map and compared to real-time ADC data
11. Runs here until external BLE control or possible stop-conditions are met.