ingenieur
wissenschaften
htw saar

STL SYSTEMTECHNIKLABOR

Embedded Systems

Autonomes Fahrzeug

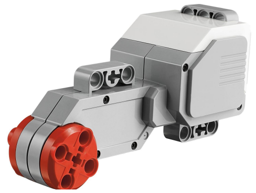
von Jens Leiner, Nicolas Loosen

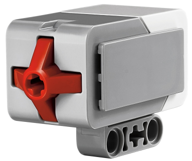
Agenda

1. Hardware
2. Demo
3. Code Aufbau
4. MISRA Besonderheiten
5. MISRA Check

Hardware

- Large Servo Motor
- Medium Servo Motor
- US-Sensor
- Touch-Sensor





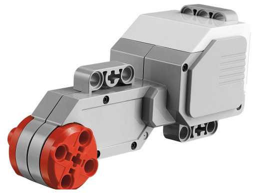
ingenieur
wissenschaften
htw saar

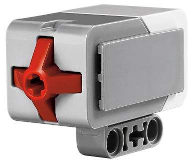
STL SYSTEMTECHNIKLABOR

Demo

Hardware

- Large Servo Motor
- Medium Servo Motor
- US-Sensor
- Touch-Sensor



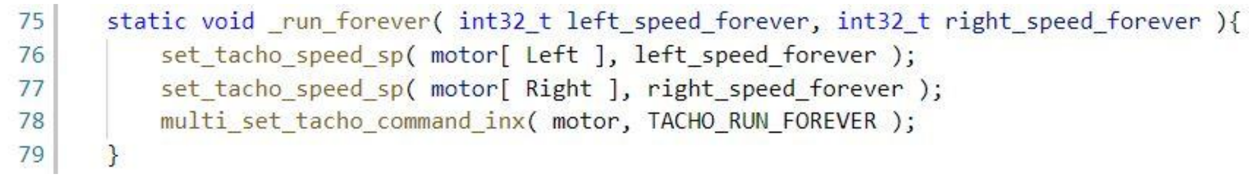


Framework ev3dev

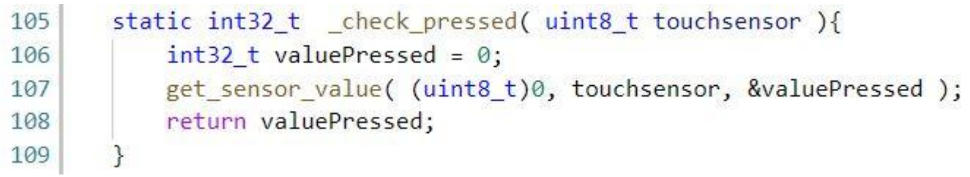
- Debian Linux based OS
- EV3, Raspberry Pi/BrickPi
- low-level driver framework
- Linux Kernel supports USB, Bluetooth devices, Wi-Fi dongle, keyboards, keypads, joysticks, cameras, SSH
- like a dual-boot on a microSD card

Quelle: https://www.ev3dev.org/

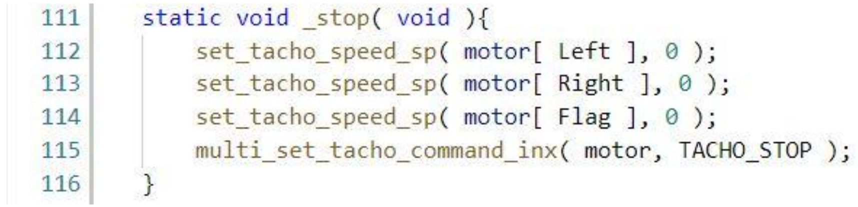
Code



```c
static void _run_forever( int32_t left_speed_forever, int32_t right_speed_forever ){
    set_tacho_speed_sp( motor[ Left ], left_speed_forever );
    set_tacho_speed_sp( motor[ Right ], right_speed_forever );
    multi_set_tacho_command_inx( motor, TACHO_RUN_FOREVER );
}
```

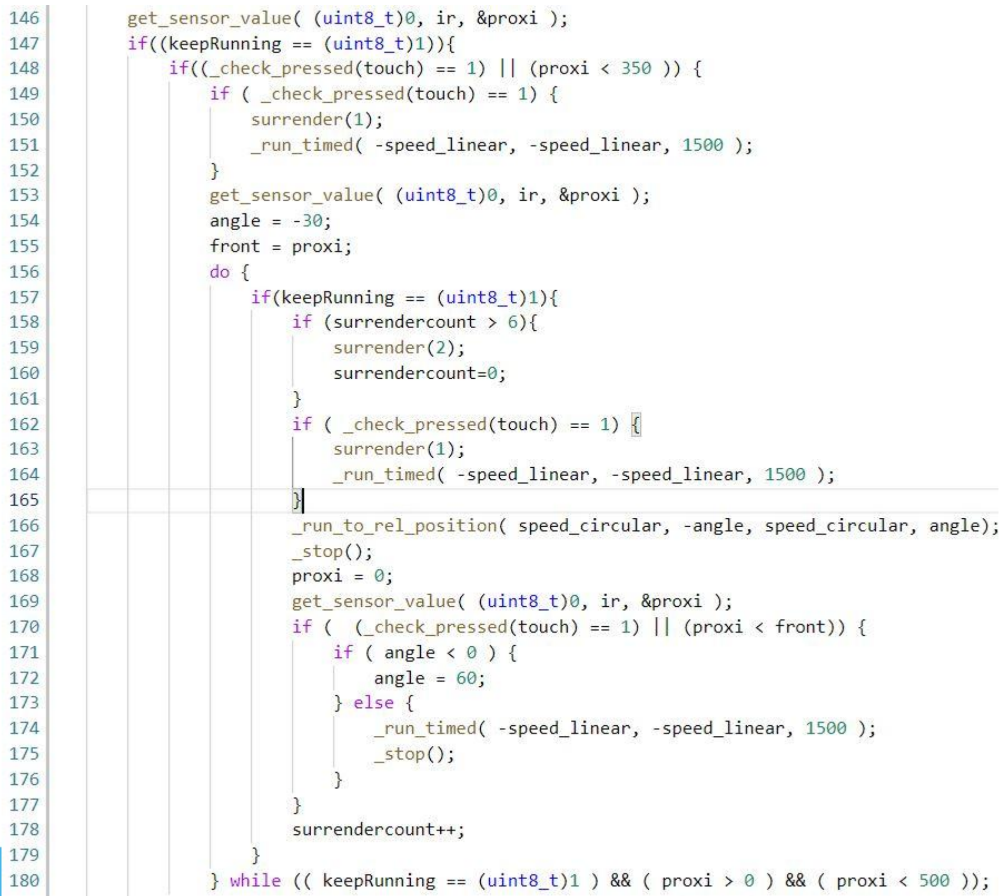


```c
static int32_t  _check_pressed( uint8_t touchsensor ){
    int32_t valuePressed = 0;
    get_sensor_value( (uint8_t)0, touchsensor, &valuePressed );
    return valuePressed;
}
```



```c
static void _stop( void ){
    set_tacho_speed_sp( motor[ Left ], 0 );
    set_tacho_speed_sp( motor[ Right ], 0 );
    set_tacho_speed_sp( motor[ Flag ], 0 );
    multi_set_tacho_command_inx( motor, TACHO_STOP );
}
```

Code

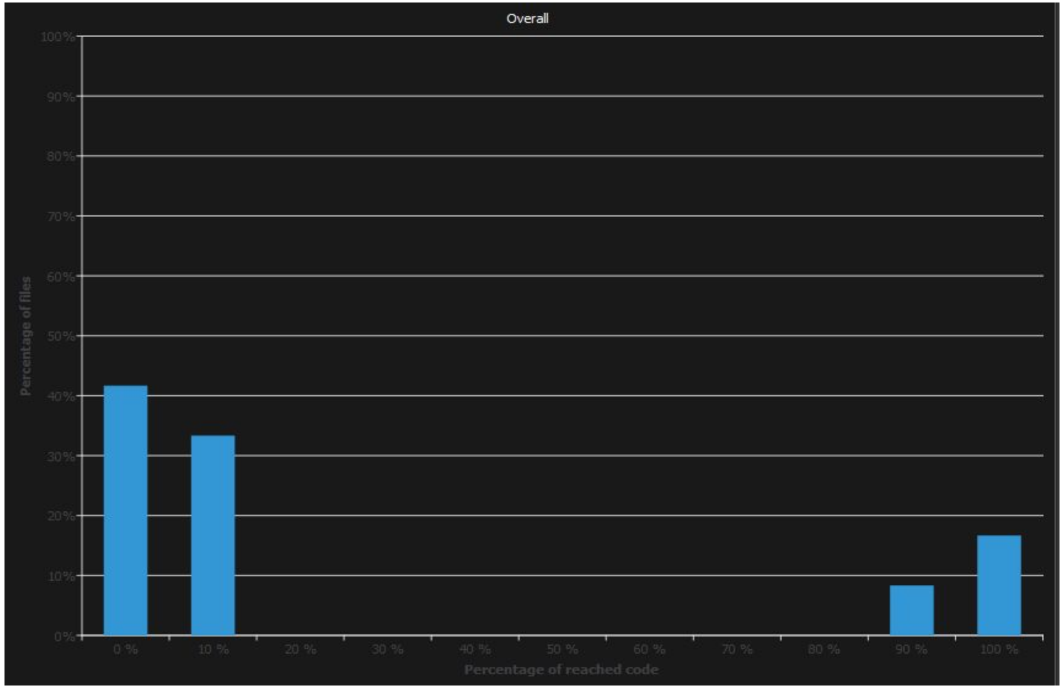


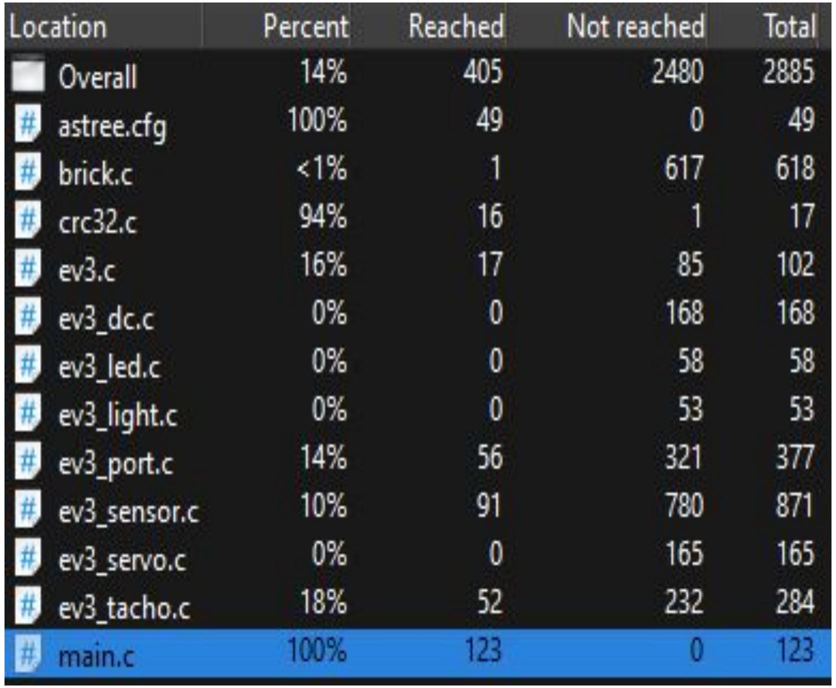
```
get_sensor_value( (uint8_t)0, ir, &proxi );
if((keepRunning == (uint8_t)1)){
    if((_check_pressed(touch) == 1) || (proxi < 350 )) {
        if ( _check_pressed(touch) == 1) {
            surrender(1);
            _run_timed( -speed_linear, -speed_linear, 1500 );
        }
        get_sensor_value( (uint8_t)0, ir, &proxi );
        angle = -30;
        front = proxi;
        do {
            if(keepRunning == (uint8_t)1){
                if (surrendercount > 6){
                    surrender(2);
                    surrendercount=0;
                }
                if ( _check_pressed(touch) == 1) {
                    surrender(1);
                    _run_timed( -speed_linear, -speed_linear, 1500 );
                }
                _run_to_rel_position( speed_circular, -angle, speed_circular, angle);
                _stop();
                proxi = 0;
                get_sensor_value( (uint8_t)0, ir, &proxi );
                if (  (_check_pressed(touch) == 1) || (proxi < front)) {
                    if ( angle < 0 ) {
                        angle = 60;
                    } else {
                        _run_timed( -speed_linear, -speed_linear, 1500 );
                        _stop();
                    }
                }
                surrendercount++;
            }
        } while (( keepRunning == (uint8_t)1 ) && ( proxi > 0 ) && ( proxi < 500 ));
```

Was musste geändert werden?

- genaue Datentypen
- Prototypen
- If-Statements
- return values
- stubs

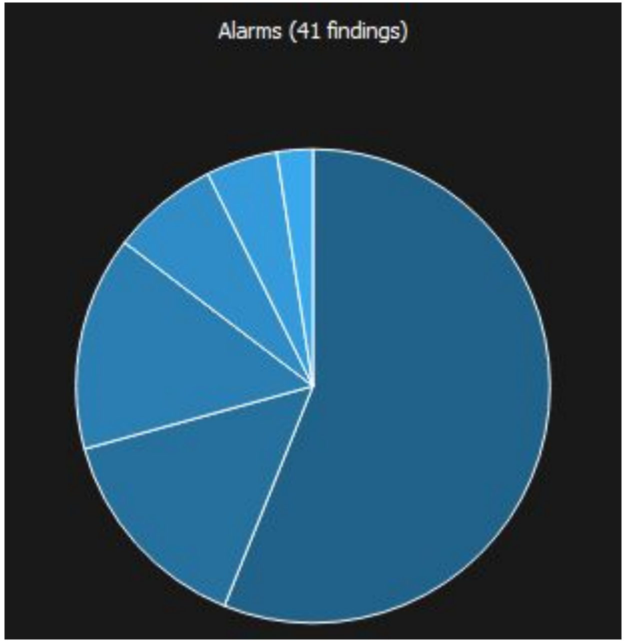
MISRA Reachability

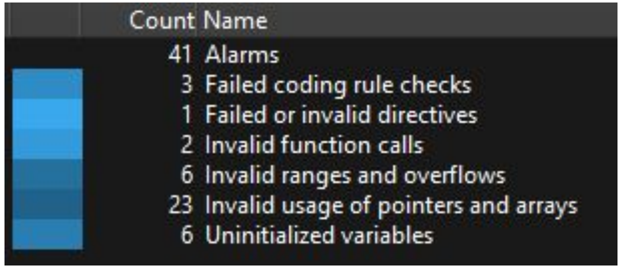




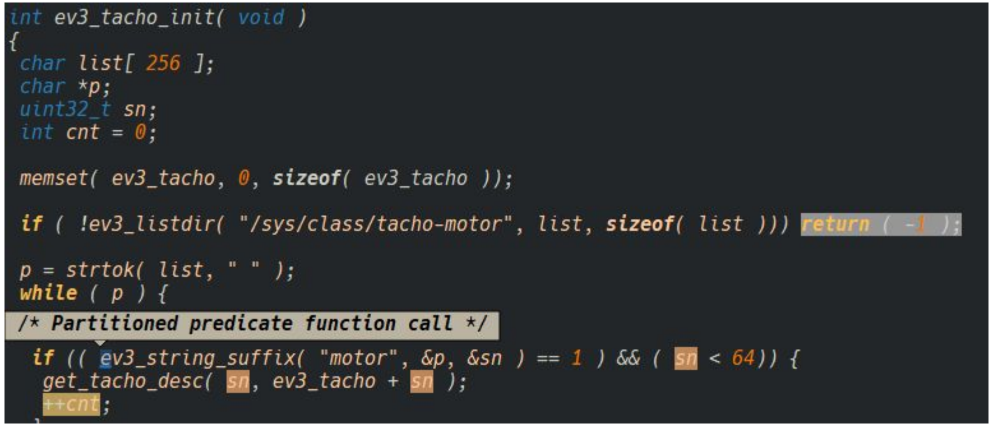
Location	Percent	Reached	Not reached	Total
Overall	14%	405	2480	2885
astree.cfg	100%	49	0	49
brick.c	<1%	1	617	618
crc32.c	94%	16	1	17
ev3.c	16%	17	85	102
ev3_dc.c	0%	0	168	168
ev3_led.c	0%	0	58	58
ev3_light.c	0%	0	53	53
ev3_port.c	14%	56	321	377
ev3_sensor.c	10%	91	780	871
ev3_servo.c	0%	0	165	165
ev3_tacho.c	18%	52	232	284
main.c	100%	123	0	123

MISRA Findings/C



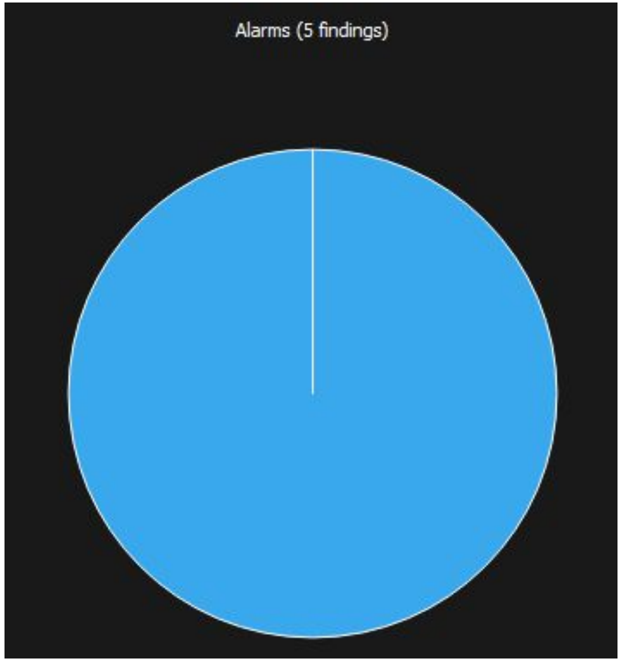


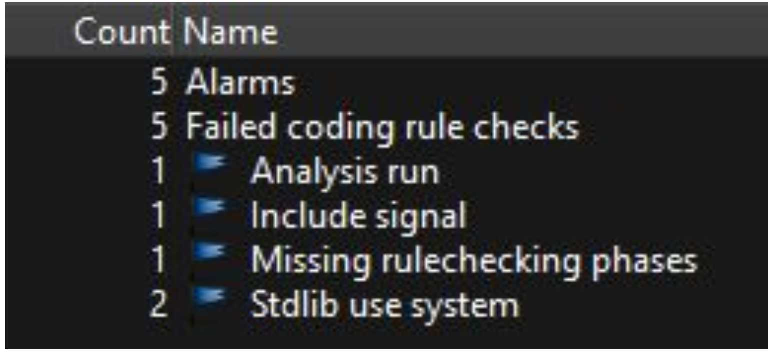
	Count	Name
	41	Alarms
🟦	3	Failed coding rule checks
🟦	1	Failed or invalid directives
🟦	2	Invalid function calls
🟦	6	Invalid ranges and overflows
🟦	23	Invalid usage of pointers and arrays
🟦	6	Uninitialized variables



```
int ev3_tacho_init( void )
{
 char list[ 256 ];
 char *p;
 uint32_t sn;
 int cnt = 0;

 memset( ev3_tacho, 0, sizeof( ev3_tacho ));

 if ( !ev3_listdir( "/sys/class/tacho-motor", list, sizeof( list ))) return ( -1 );

 p = strtok( list, " " );
 while ( p ) {
/* Partitioned predicate function call */
  if (( ev3_string_suffix( "motor", &p, &sn ) == 1 ) && ( sn < 64)) {
   get_tacho_desc( sn, ev3_tacho + sn );
   ++cnt;
```

MISRA Rule Violations

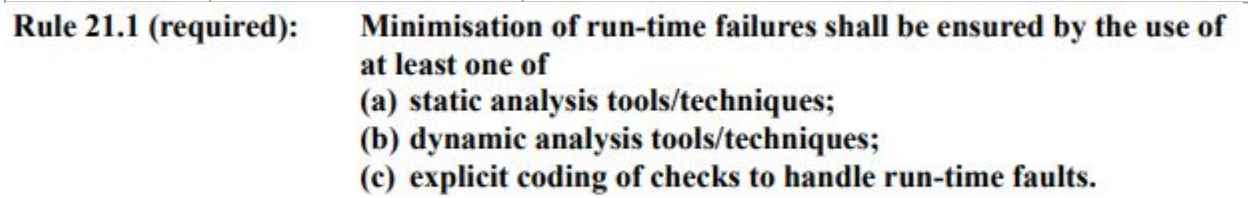




Count	Name
5	Alarms
5	Failed coding rule checks
1	Analysis run
1	Include signal
1	Missing rulechecking phases
2	Stdlib use system

MISRA Rule Violations

Type	Category	Message
Alarm (R)	Analysis run	ALARM (R) check_analysis_run: check failed (violates M.21.1-required)
Alarm (R)	Missing rulechecking Phase	ALARM (R) check_missing_rulechecking_phases: check failed (violates A.5.4)

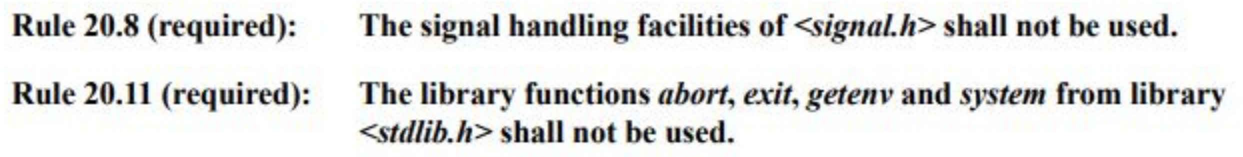


Rule 21.1 (required): **Minimisation of run-time failures shall be ensured by the use of at least one of**
(a) static analysis tools/techniques;
(b) dynamic analysis tools/techniques;
(c) explicit coding of checks to handle run-time faults.

ingenieur
wissenschaften
htw saar

MISRA Rule Violations

Type	Category	Message
Alarm (R)	stdlib use system	ALARM (R) check_stdlib_use_system: check failed (violates M.20.11-required)
Alarm (R)	Include Signal	ALARM (R) check_include_signal: check failed (violates M.20.8-required)



Rule 20.8 (required):	**The signal handling facilities of <*signal.h*> shall not be used.**
Rule 20.11 (required):	**The library functions *abort*, *exit*, *getenv* and *system* from library <*stdlib.h*> shall not be used.**

Vielen Dank für die Aufmerksamkeit!