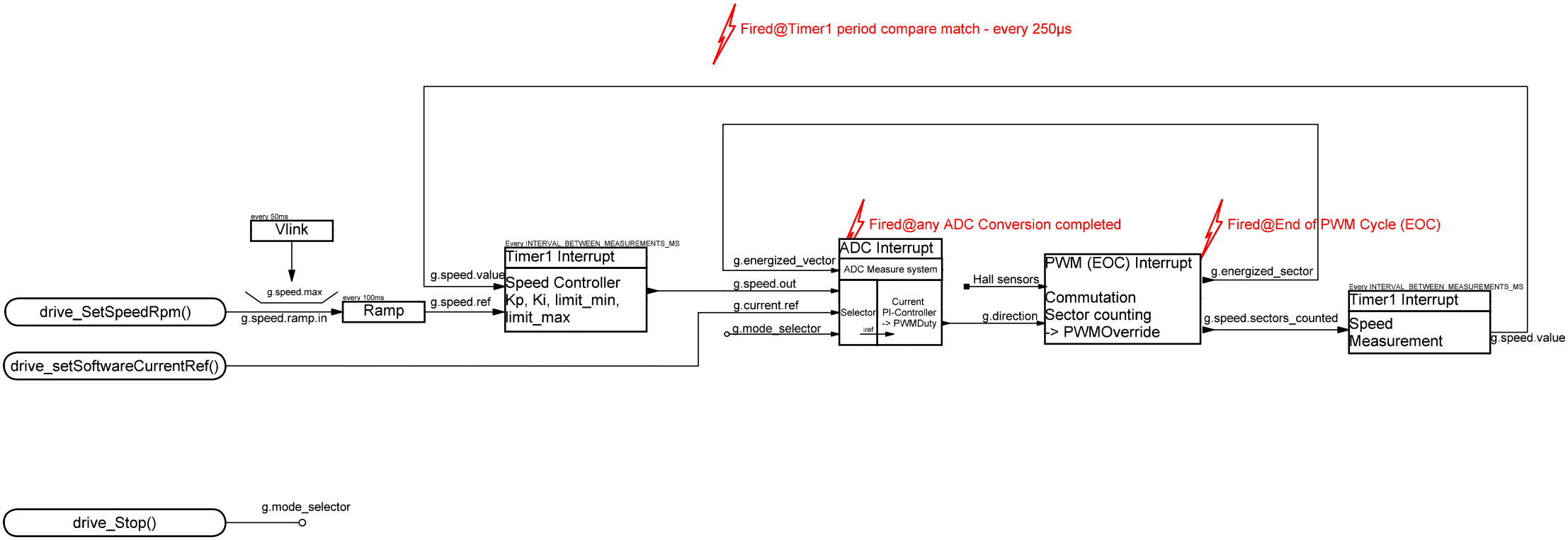
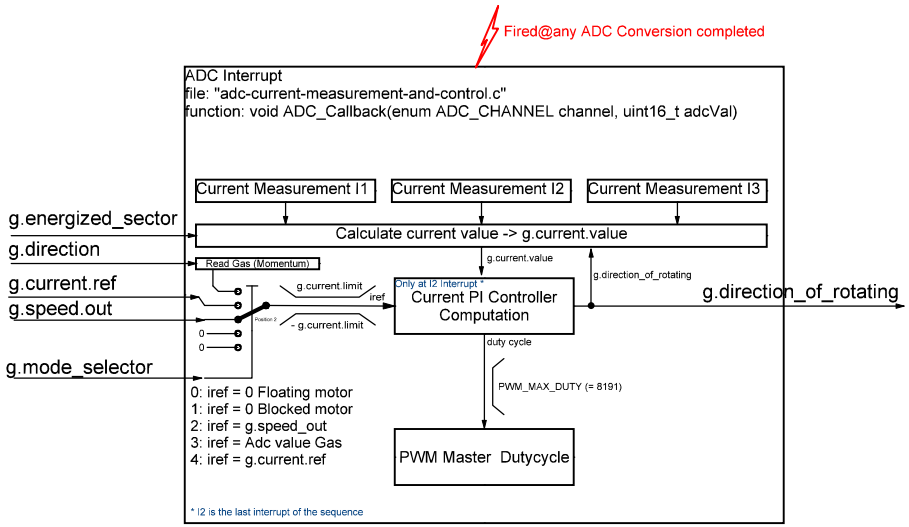
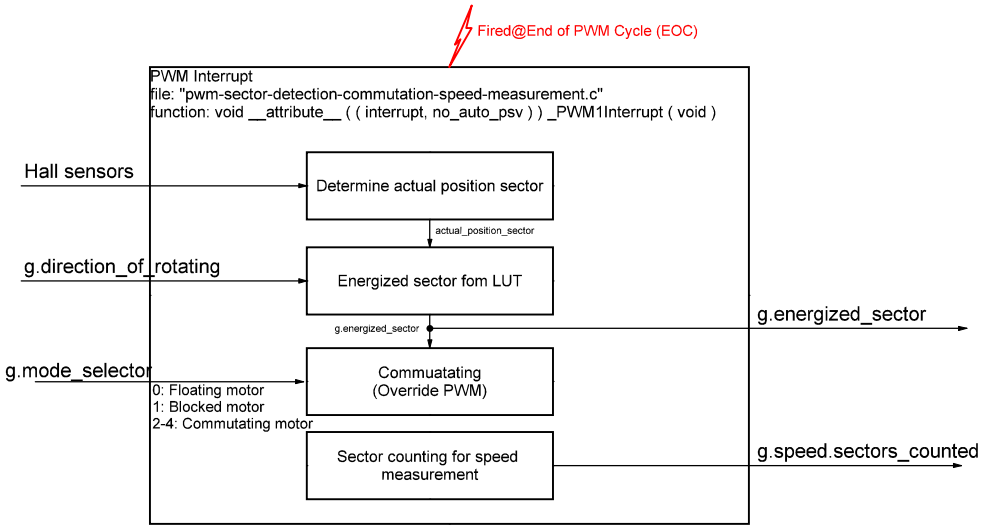
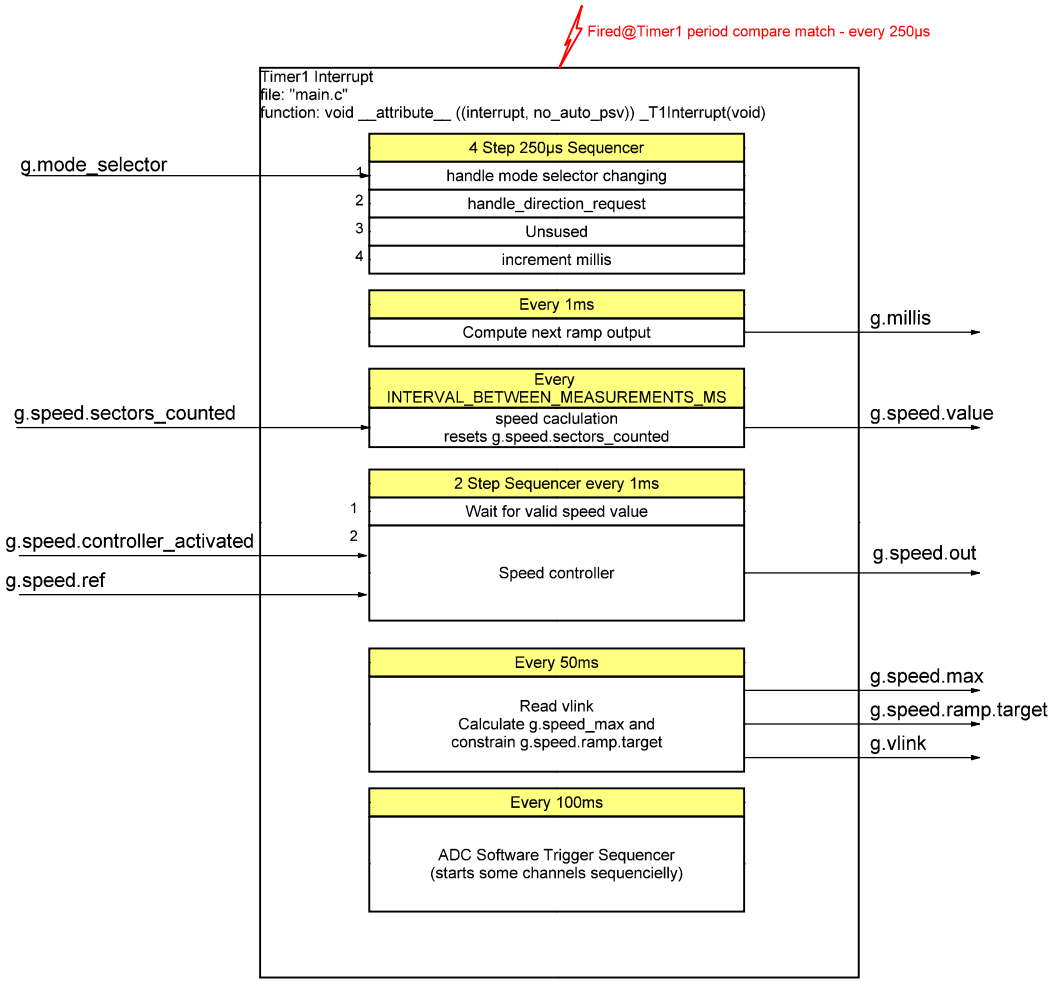

Fired@Timer1 period compare match - every 250µs
every 50ms
Vlink
g.speed.max
drive_SetSpeedRpm()
g.speed.ramp.in
every 100ms
Ramp
g.speed.ref
g.speed.value
Every INTERVAL_BETWEEN_MEASUREMENTS_MS
Timer1 Interrupt
Speed Controller
Kp, Ki, limit_min,
limit_max
g.speed.out
g.current.ref
g.mode_selector
g.energized_vector
Fired@any ADC Conversion completed
ADC Interrupt
ADC Measure system
Selector
iref
Current
PI-Controller
-> PWMDuty
Hall sensors
g.direction
PWM (EOC) Interrupt
Commutation
Sector counting
-> PWMOverride
Fired@End of PWM Cycle (EOC)
g.energized_sector
g.speed.sectors_counted
Every INTERVAL_BETWEEN_MEASUREMENTS_MS
Timer1 Interrupt
Speed
Measurement
g.speed.value
drive_setSoftwareCurrentRef()
drive_Stop()
g.mode_selector


Fired@any ADC Conversion completed
ADC Interrupt
file: "adc-current-measurement-and-control.c"
function: void ADC_Callback(enum ADC_CHANNEL channel, uint16_t adcVal)
Current Measurement I1
Current Measurement I2
Current Measurement I3
Calculate current value -> g.current.value
g.energized_sector
g.direction
g.current.ref
g.speed.out
g.mode_selector
Read Gas (Momentum)
g.current.value
g.direction_of_rotating
g.current.limit
iref
- g.current.limit
Only at I2 Interrupt *
Current PI Controller Computation
g.direction_of_rotating
duty cycle
PWM_MAX_DUTY (= 8191)
PWM Master Dutycycle
0: iref = 0 Floating motor
1: iref = 0 Blocked motor
2: iref = g.speed_out
3: iref = Adc value Gas
4: iref = g.current.ref
* I2 is the last interrupt of the sequence


Fired@End of PWM Cycle (EOC)
PWM Interrupt
file: "pwm-sector-detection-commutation-speed-measurement.c"
function: void __attribute__ ((interrupt, no_auto_psv)) _PWM1Interrupt (void)
Hall sensors
Determine actual position sector
actual_position_sector
g.direction_of_rotating
Energized sector fom LUT
g.energized_sector
g.energized_sector
g.mode_selector
Commuatating (Override PWM)
0: Floating motor
1: Blocked motor
2-4: Commutating motor
Sector counting for speed measurement
g.speed.sectors_counted


Fired@Timer1 period compare match - every 250µs
Timer1 Interrupt
file: "main.c"
function: void __attribute__ ((interrupt, no_auto_psv)) _T1Interrupt(void)
g.mode_selector
4 Step 250µs Sequencer
1 handle mode selector changing
2 handle_direction_request
3 Unsused
4 increment millis
Every 1ms
Compute next ramp output
g.millis
g.speed.sectors_counted
Every INTERVAL_BETWEEN_MEASUREMENTS_MS
speed caclulation
resets g.speed.sectors_counted
g.speed.value
2 Step Sequencer every 1ms
1 Wait for valid speed value
g.speed.controller_activated
2
g.speed.ref
Speed controller
g.speed.out
Every 50ms
Read vlink
Calculate g.speed_max and
constrain g.speed.ramp.target
g.speed.max
g.speed.ramp.target
g.vlink
Every 100ms
ADC Software Trigger Sequencer
(starts some channels sequencielly)

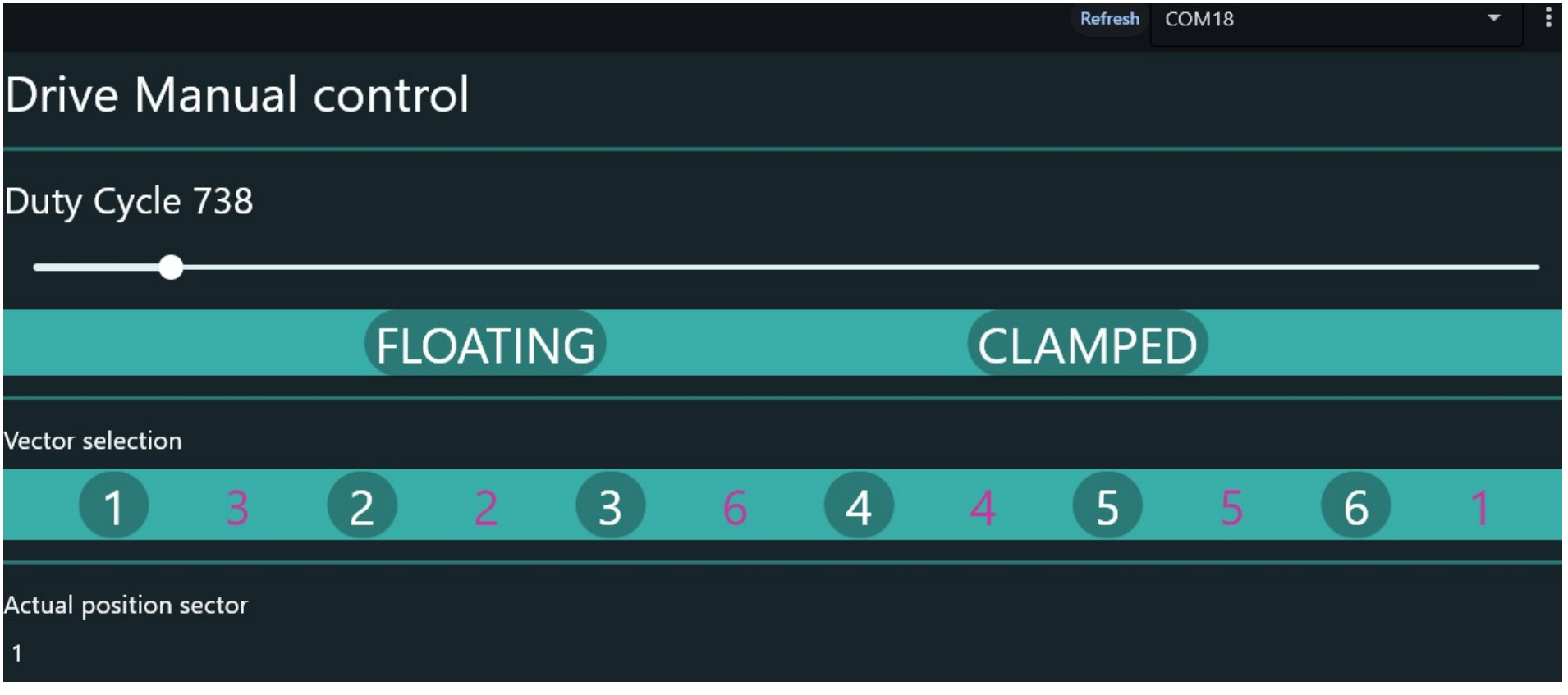
defined in global.h

```
#define MODE_SELECTOR_ZERO_MOTOR_BLOCKED 0
#define MODE_SELECTOR_ZERO_MOTOR_FLOATING 1
#define MODE_SELECTOR_SPEEDCONTROLLER 2
#define MODE_SELECTOR_MOMENTUM 3
#define MODE_SELECTOR_IREF 4
```

Package:	TQFP48	Pin No:	8	9	10	11	12	21	22	25	26	27	33	34	35	36	37	45	46	47	48	1	2	7	15	16	20	38	39	17	24	28	29	40	41	3	4	44	30	23	6
Module	Function	Direction	PORTA					PORTB																PORTC															PORTD			
All			0	1	2	3	4	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	0	1	2	3	4	5	6	7	8	9	10	11	12	13	1	8	10	13
ADC1	ANx	input																																							
	ADCTRG31	input																																							
Clock	CLKO	output																																							
	REFI	input																																							
	REFO	output																																							
ICD	PGCx	input																																							
	PGDx	input																																							
Pins	GPIO	input																																							
	GPIO	output																																							

Channel	Pins	Enable	Custom Name	Trigger Source	Interrupts S
AN0	RA0	☑	_I1	PWM Generator1 ADC Trigger1	☑
AN1	RB2	☑	_TEMPERA	None	☐
AN2	RB7	☐	Channel_A	None	☐
AN3	RA3	☑	_I3	PWM Generator1 ADC Trigger1	☑
AN4	RA4	☑	_I2	PWM Generator1 ADC Trigger1	☑
AN5	RB0	☐	Channel_A	CLC2 Output	☐
AN6	RB1	☑	_MOMENT	None	☐
AN7	RB2	☐		None	☐
AN8	RB3	☐	Channel_A	None	☐
AN9	RA2	☑	_TEMPERA	None	☐
AN10	RB8	☐	Channel_A	None	☐
AN11	RB9	☐		None	☐
AN12	RC0	☑	_VLINK	Common Software Trigger	☐
AN13	RC1	☑	_UM3	None	☐
AN14	RC2	☑	_UM2	None	☐
AN15	RC3	☑	_UM1	None	☐
AN16	Internal-Te	☐	Channel_A	None	☐
AN17	Internal-Ba	☐	Channel_A	None	☐

```
18
19  // #define DEBUG                // only one option (DEBUG or FLETUINO) is allowed
20  // #define DEBUG_SERIAL_COMMAND_HANDLING
21  #define FLETUINO                // DEBUG or FLETUINO or nothing
22  // #define FLETUINO_APPLICATION_DEMO
23  // #define FLETUINO_PI_CONTROLLER_SETTINGS
24  #define FLETUINO_MANUAL_CONTROL
25
```



```
19  // #define DEBUG                    // only one option (DEBUG or FLETUINO) is allowed
20  // #define DEBUG_SERIAL_COMMAND_HANDLING
21  #define FLETUINO                    // DEBUG or FLETUINO or nothing
22  // #define FLETUINO_APPLICATION_DEMO
23  #define FLETUINO_PI_CONTROLLER_SETTINGS
24  // #define FLETUINO_MANUAL_CONTROL
```

```
19  // #define DEBUG                  // only one option (DEBUG or FLETUINO) is allowed
20  // #define DEBUG_SERIAL_COMMAND_HANDLING
21  #define FLETUINO                  // DEBUG or FLETUINO or nothing
22  #define FLETUINO_APPLICATION_DEMO
23  // #define FLETUINO_PI_CONTROLLER_SETTINGS
24  // #define FLETUINO_MANUAL_CONTROL
25
```