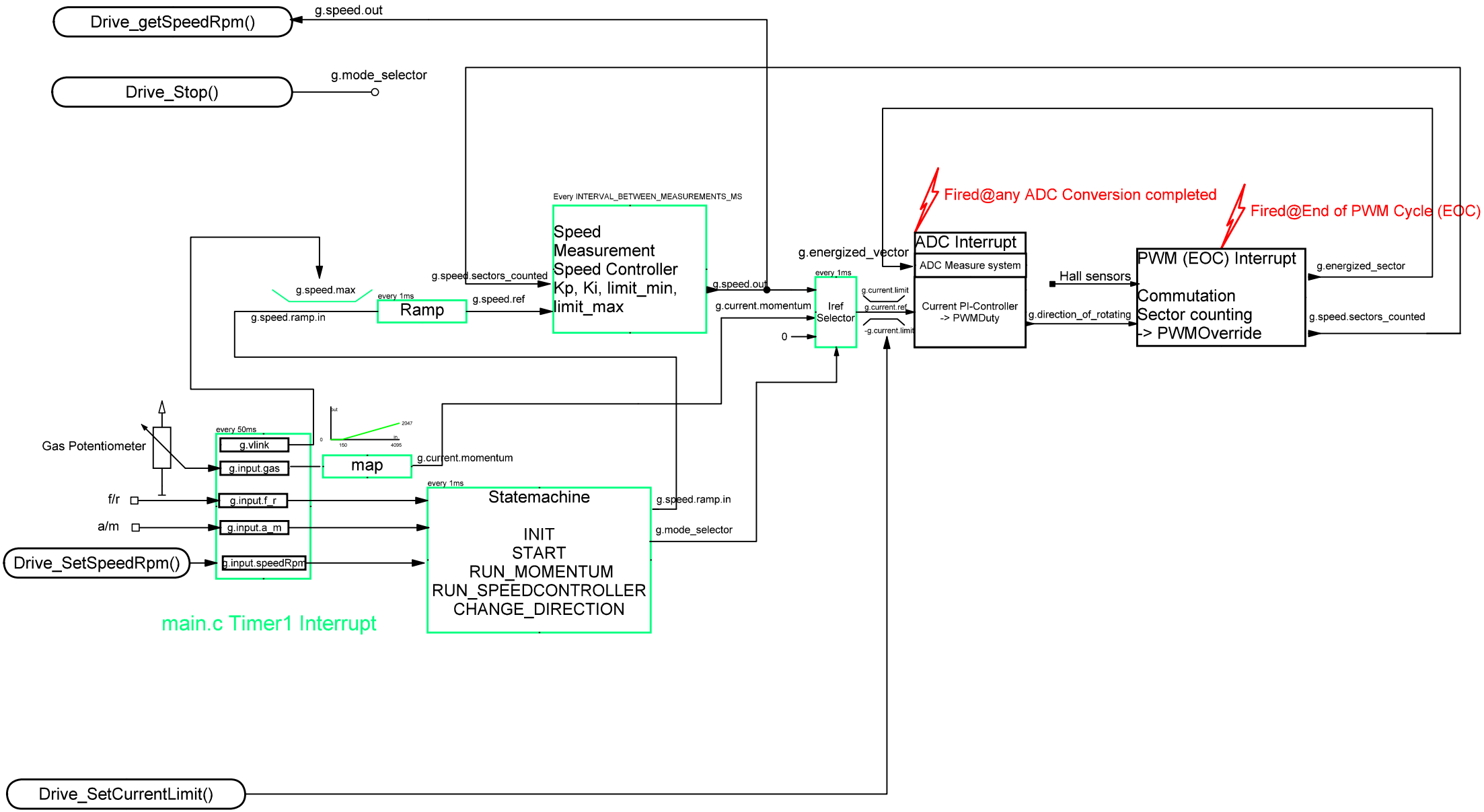
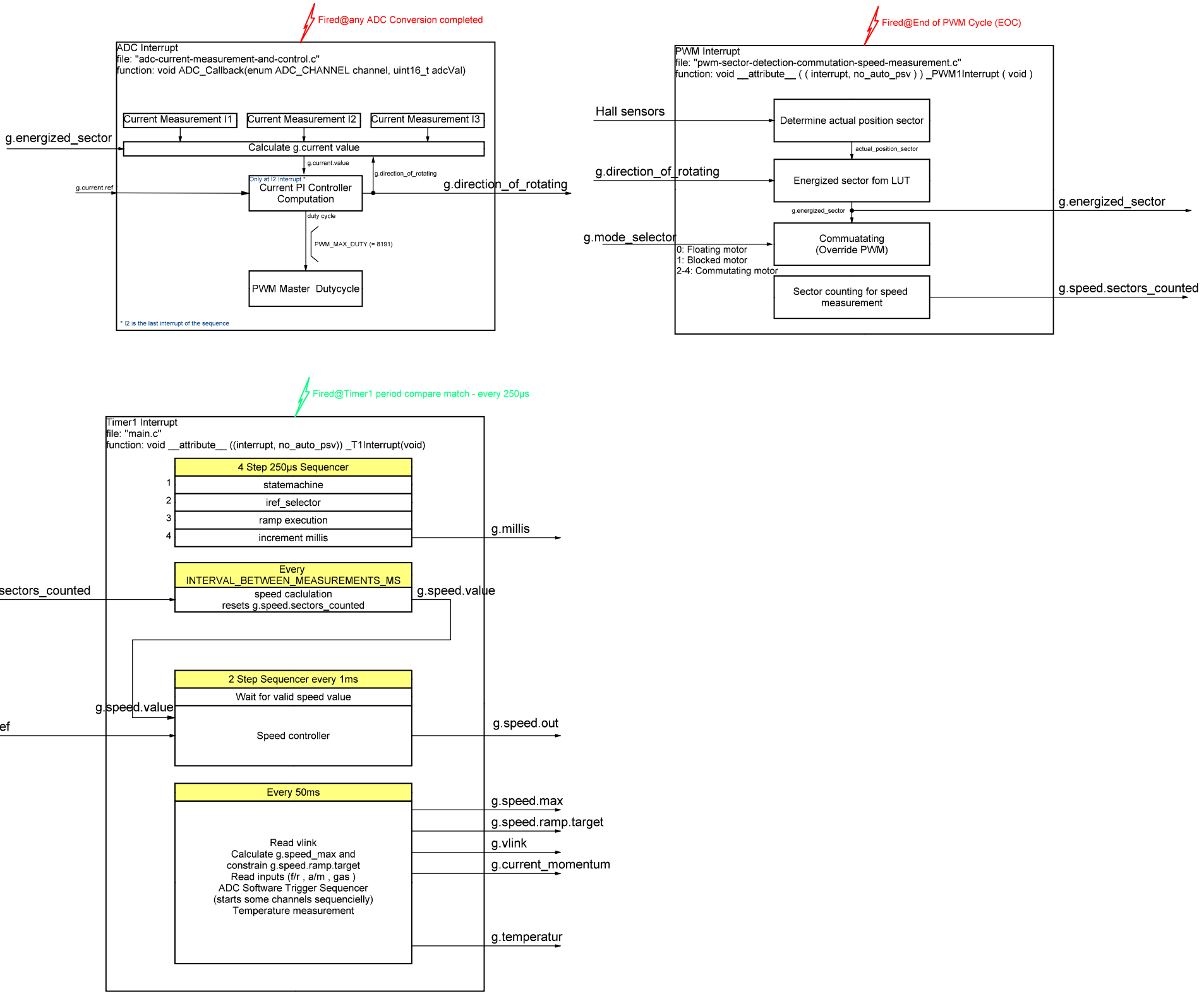
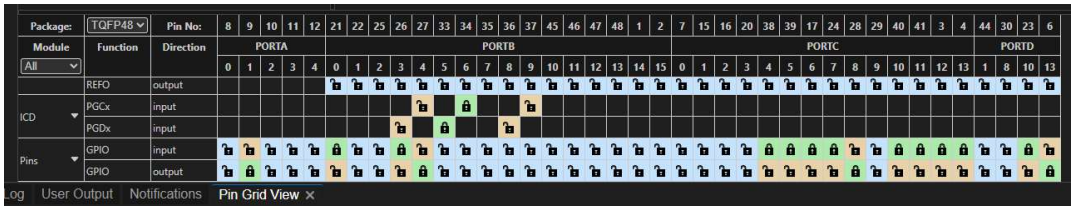
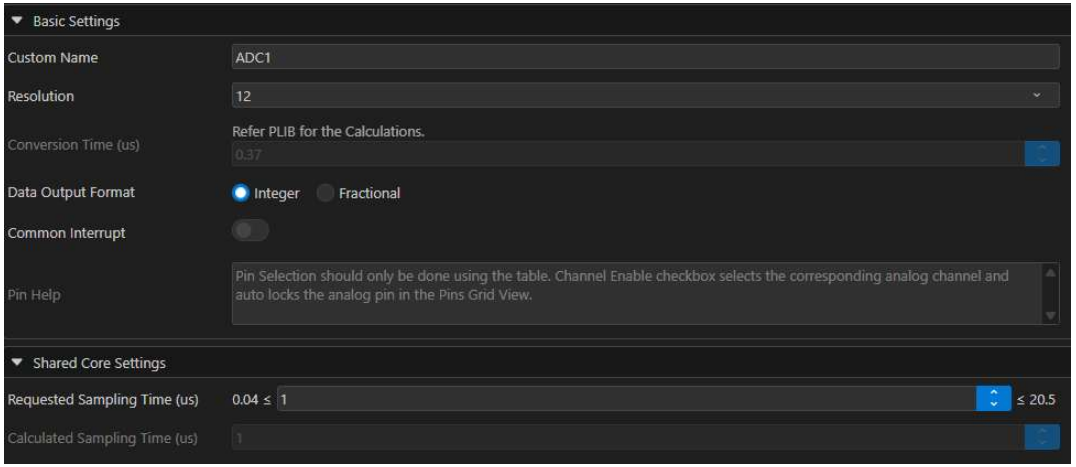

Drive_getSpeedRpm()
g.speed.out
Drive_Stop()
g.mode_selector
Every INTERVAL_BETWEEN_MEASUREMENTS_MS
Speed Measurement
Speed Controller
Kp, Ki, limit_min, limit_max
g.speed.sectors_counted
g.speed.max
every 1ms
Ramp
g.speed.ref
g.speed.ramp.in
g.speed.out
g.current.momentum
0
every 1ms
Iref Selector
g.current.limit
g.current.ref
-g.current.limit
g.energized_vector
Fired@any ADC Conversion completed
ADC Interrupt
ADC Measure system
Current PI-Controller
-> PWMDuty
g.direction_of_rotating
Hall sensors
Fired@End of PWM Cycle (EOC)
PWM (EOC) Interrupt
Commutation
Sector counting
-> PWMOverride
g.energized_sector
g.speed.sectors_counted
Gas Potentiometer
every 50ms
g.vlink
g.input.gas
map
g.current.momentum
f/r
g.input.f_r
a/m
g.input.a_m
Drive_SetSpeedRpm()
g.input.speedRpm
every 1ms
Statemachine
INIT
START
RUN_MOMENTUM
RUN_SPEEDCONTROLLER
CHANGE_DIRECTION
g.speed.ramp.in
g.mode_selector
main.c Timer1 Interrupt
Drive_SetCurrentLimit()


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
g.energized_sector
Calculate g.current value
g.current.value
g.direction_of_rotating
g.current.ref
Only at I2 Interrupt *
Current PI Controller Computation
g.direction_of_rotating
duty cycle
PWM_MAX_DUTY (= 8191)
PWM Master Dutycycle
* 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)
4 Step 250µs Sequencer
1 statemachine
2 iref_selector
3 ramp execution
4 increment millis
g.millis
Every INTERVAL_BETWEEN_MEASUREMENTS_MS
speed caclulation
resets g.speed.sectors_counted
g.speed.sectors_counted
g.speed.value
2 Step Sequencer every 1ms
Wait for valid speed value
g.speed.value
g.speed.ref
Speed controller
g.speed.out
Every 50ms
Read vlink
Calculate g.speed_max and constrain g.speed.ramp.target
Read inputs (f/r , a/m , gas)
ADC Software Trigger Sequencer
(starts some channels sequencielly)
Temperature measurement
g.speed.max
g.speed.ramp.target
g.vlink
g.current_momentum
g.temperatur

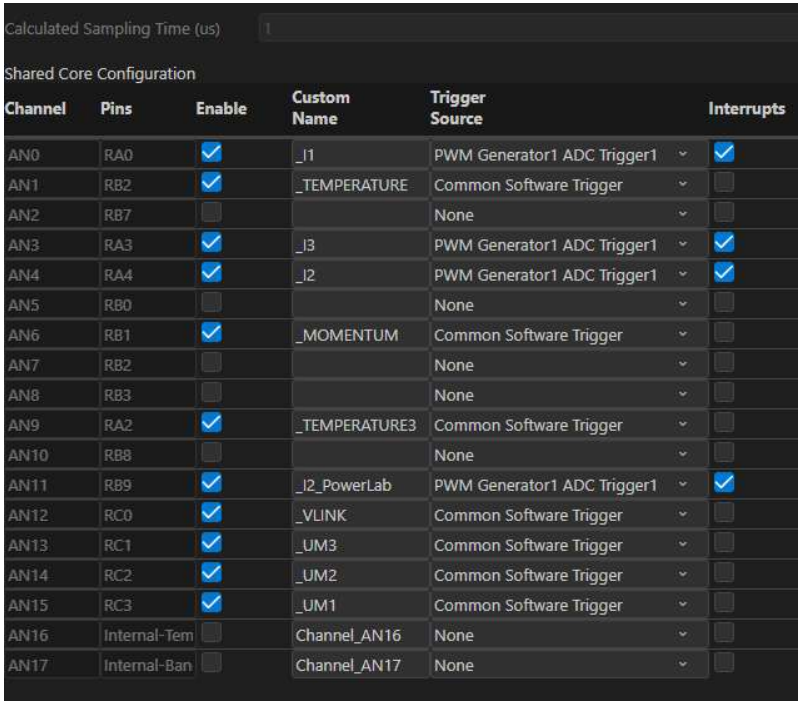
Pin Grid



ADC Basic

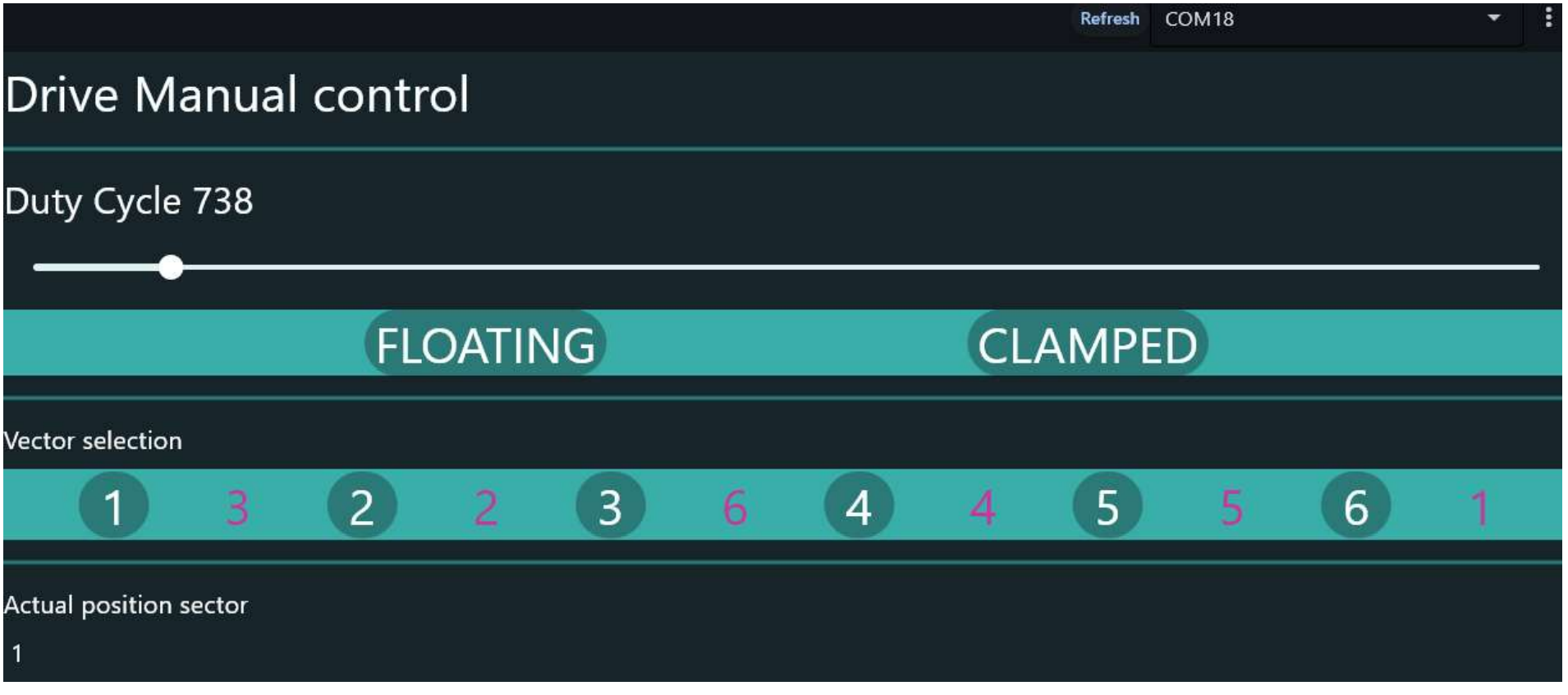


ADC Shared Core Settings

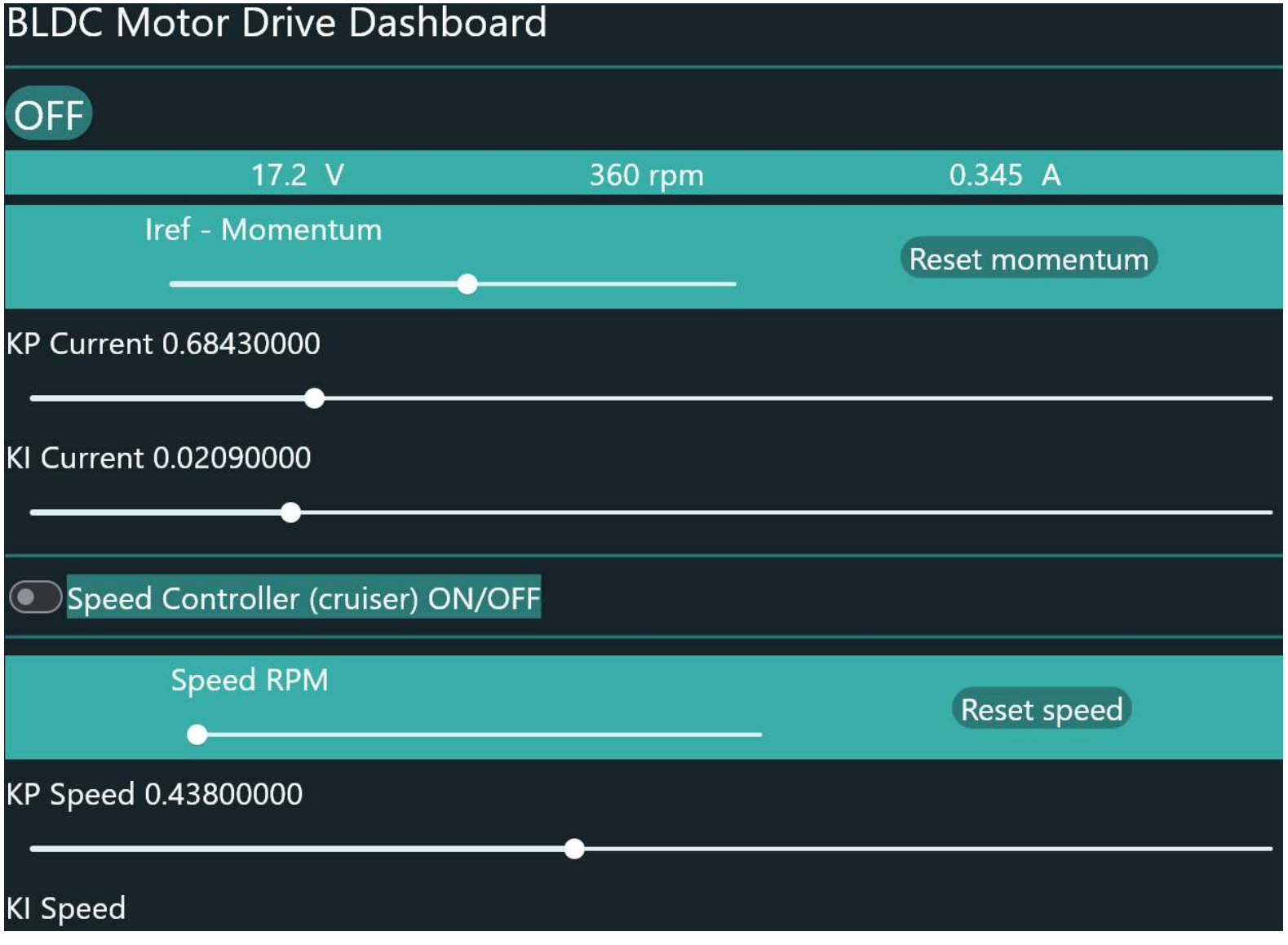
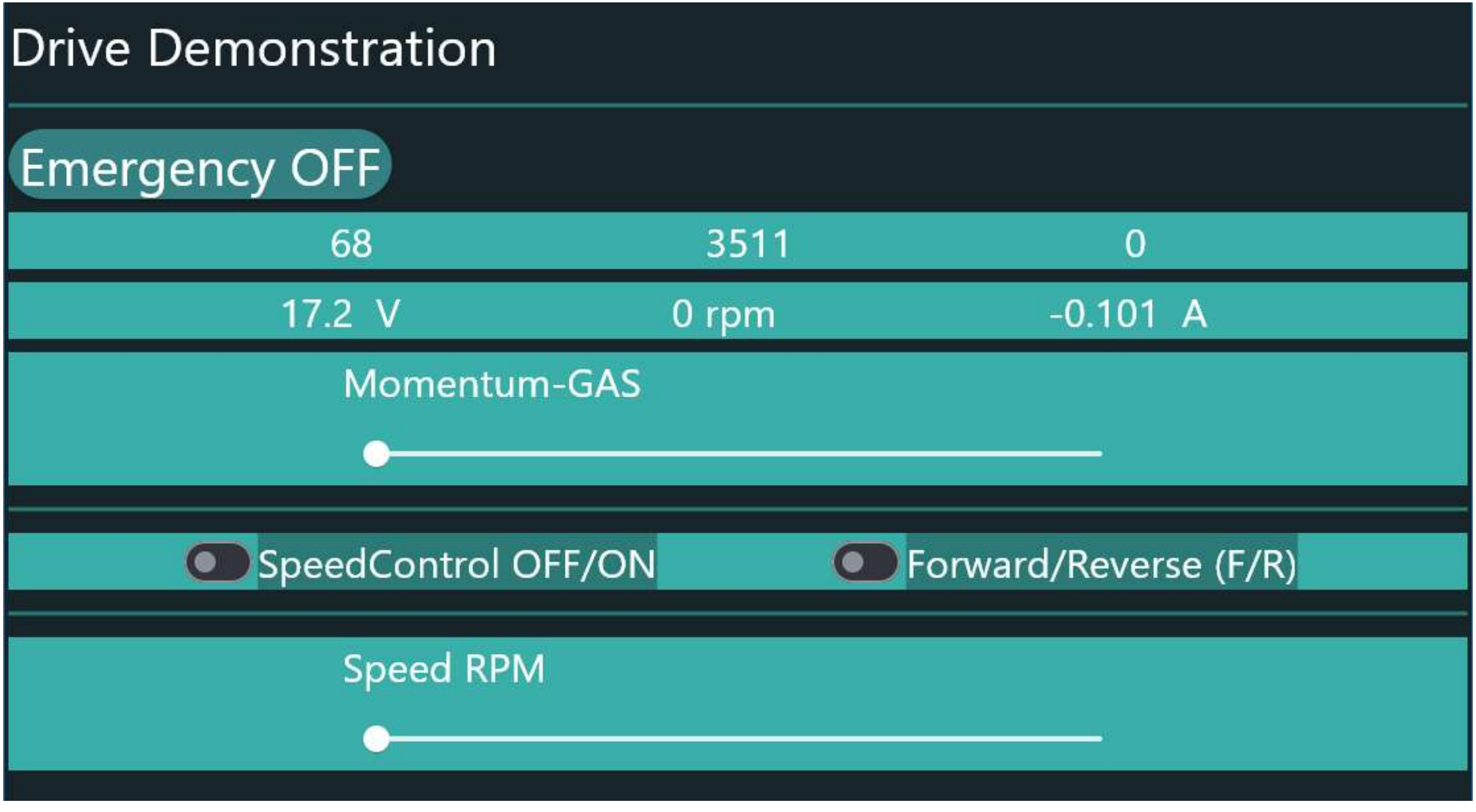


Channel	Pins	Enable	Custom Name	Trigger Source	Interrupts
AN0	RA0	☑	_I1	PWM Generator1 ADC Trigger1	☑
AN1	RB2	☑	_TEMPERATURE	Common Software Trigger	☐
AN2	RB7	☐		None	☐
AN3	RA3	☑	_I3	PWM Generator1 ADC Trigger1	☑
AN4	RA4	☑	_I2	PWM Generator1 ADC Trigger1	☑
AN5	RB0	☐		None	☐
AN6	RB1	☑	_MOMENTUM	Common Software Trigger	☐
AN7	RB2	☐		None	☐
AN8	RB3	☐		None	☐
AN9	RA2	☑	_TEMPERATURE3	Common Software Trigger	☐
AN10	RB8	☐		None	☐
AN11	RB9	☑	_I2_PowerLab	PWM Generator1 ADC Trigger1	☑
AN12	RC0	☑	_VLINK	Common Software Trigger	☐
AN13	RC1	☑	_UM3	Common Software Trigger	☐
AN14	RC2	☑	_UM2	Common Software Trigger	☐
AN15	RC3	☑	_UM1	Common Software Trigger	☐
AN16	Internal-Tem	☐	Channel_AN16	None	☐
AN17	Internal-Ban	☐	Channel_AN17	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
```