

Seguidor

100.1

Generated by Doxygen 1.8.16



<b>1 Hierarchical Index</b>	<b>1</b>
1.1 Class Hierarchy	1
<b>2 Class Index</b>	<b>3</b>
2.1 Class List	3
<b>3 File Index</b>	<b>5</b>
3.1 File List	5
<b>4 Class Documentation</b>	<b>7</b>
4.1 ADC Class Reference	7
4.1.1 Constructor & Destructor Documentation	7
4.1.1.1 ADC()	7
4.1.2 Member Function Documentation	8
4.1.2.1 analogRead()	8
4.1.2.2 ConfigADCPin()	8
4.1.2.3 GetADCChannel()	8
4.1.2.4 SetADCChannel()	8
4.2 ADC_TypeDef Struct Reference	8
4.2.1 Member Data Documentation	9
4.2.1.1 CR1	9
4.2.1.2 CR2	9
4.2.1.3 DR	9
4.2.1.4 HTR	9
4.2.1.5 JDR1	10
4.2.1.6 JDR2	10
4.2.1.7 JDR3	10
4.2.1.8 JDR4	10
4.2.1.9 JOFR1	10
4.2.1.10 JOFR2	10
4.2.1.11 JOFR3	10
4.2.1.12 JOFR4	10
4.2.1.13 JSQR	11
4.2.1.14 LTR	11
4.2.1.15 SMPR1	11
4.2.1.16 SMPR2	11
4.2.1.17 SQR1	11
4.2.1.18 SQR2	11
4.2.1.19 SQR3	11
4.2.1.20 SR	12
4.3 AFIO_TypeDef Struct Reference	12
4.3.1 Member Data Documentation	12
4.3.1.1 EVCR	12

4.3.1.2 EXTICR . . . . .	12
4.3.1.3 MAPR . . . . .	12
4.3.1.4 MAPR2 . . . . .	13
4.3.1.5 RESERVED0 . . . . .	13
4.4 ANGLE_LINE_SENSOR_H Class Reference . . . . .	13
4.4.1 Member Function Documentation . . . . .	13
4.4.1.1 readAngle() . . . . .	13
4.4.1.2 updateLineAngle() . . . . .	13
4.5 Communication Class Reference . . . . .	14
4.5.1 Constructor & Destructor Documentation . . . . .	14
4.5.1.1 Communication() . . . . .	14
4.5.2 Member Function Documentation . . . . .	14
4.5.2.1 HandlerByTime() . . . . .	14
4.5.2.2 verifyCommand() . . . . .	14
4.5.2.3 waitForCommand() . . . . .	15
4.5.3 Member Data Documentation . . . . .	15
4.5.3.1 command . . . . .	15
4.6 Controller Class Reference . . . . .	15
4.6.1 Constructor & Destructor Documentation . . . . .	15
4.6.1.1 Controller() . . . . .	16
4.6.2 Member Function Documentation . . . . .	16
4.6.2.1 controlRule() . . . . .	16
4.6.2.2 getVcontrol() . . . . .	16
4.6.2.3 getWcontrol() . . . . .	16
4.6.2.4 HandlerByTime() . . . . .	16
4.6.2.5 reset() . . . . .	16
4.6.2.6 setSpeedRef() . . . . .	16
4.6.2.7 start() . . . . .	17
4.6.2.8 stop() . . . . .	17
4.6.3 Member Data Documentation . . . . .	17
4.6.3.1 Robot . . . . .	17
4.7 Encoder Class Reference . . . . .	17
4.7.1 Constructor & Destructor Documentation . . . . .	18
4.7.1.1 Encoder() [1/2] . . . . .	18
4.7.1.2 Encoder() [2/2] . . . . .	18
4.7.2 Member Function Documentation . . . . .	18
4.7.2.1 Encoder_Handler() . . . . .	18
4.7.2.2 Encoder_Handler_by_Time() . . . . .	18
4.7.2.3 Encoder_Initiallize() . . . . .	18
4.7.2.4 getDeltaTicks() . . . . .	19
4.7.2.5 getDeltaTime() . . . . .	19
4.7.2.6 getDirection() . . . . .	19

4.7.2.7 getLastSpeed()	19
4.7.2.8 getLastTicksTime()	19
4.7.2.9 getNotFilteredSpeed()	19
4.7.2.10 getSpeed()	19
4.7.2.11 getTeta()	19
4.7.2.12 getTicks()	20
4.7.2.13 getTicksTime()	20
4.7.2.14 reset()	20
4.8 FLASH_TypeDef Struct Reference	20
4.8.1 Member Data Documentation	20
4.8.1.1 ACR	20
4.8.1.2 AR	21
4.8.1.3 CR	21
4.8.1.4 KEYR	21
4.8.1.5 OBR	21
4.8.1.6 OPTKEYR	21
4.8.1.7 RESERVED	21
4.8.1.8 SR	21
4.8.1.9 WRPR	22
4.9 GPIO Class Reference	22
4.9.1 Constructor & Destructor Documentation	22
4.9.1.1 GPIO() [1/3]	22
4.9.1.2 GPIO() [2/3]	23
4.9.1.3 GPIO() [3/3]	23
4.9.2 Member Function Documentation	23
4.9.2.1 Config_PU_PD()	23
4.9.2.2 ConfigGPIOPin()	23
4.9.2.3 digitalRead()	24
4.9.2.4 digitalWrite()	24
4.9.2.5 GetGPIOMode()	24
4.9.2.6 GetGPIOPinNumber()	24
4.9.2.7 GetGPIOPort()	24
4.9.2.8 GetGPIOPuPd()	24
4.9.2.9 GetGPIOState()	24
4.9.2.10 GetIOPin()	25
4.9.2.11 SetGPIOMode()	25
4.9.2.12 SetGPIOPortPin()	25
4.9.2.13 SetIOPin()	25
4.9.2.14 togglePin()	25
4.10 GPIO_STRUCT Struct Reference	25
4.10.1 Member Data Documentation	26
4.10.1.1 pinNumber	26

4.10.1.2 Port	26
4.11 GPIO_TypeDef Struct Reference	26
4.11.1 Member Data Documentation	26
4.11.1.1 BRR	26
4.11.1.2 BSRR	27
4.11.1.3 CRH	27
4.11.1.4 CRL	27
4.11.1.5 IDR	27
4.11.1.6 LCKR	27
4.11.1.7 ODR	27
4.12 Kinematic Class Reference	27
4.12.1 Constructor & Destructor Documentation	28
4.12.1.1 Kinematic() [1/2]	29
4.12.1.2 Kinematic() [2/2]	29
4.12.2 Member Function Documentation	29
4.12.2.1 calibrateLineSensor()	29
4.12.2.2 getLineAngle()	29
4.12.2.3 getLineAngleNotFiltered()	29
4.12.2.4 getLinePosition()	29
4.12.2.5 getTeta()	30
4.12.2.6 getV()	30
4.12.2.7 getW()	30
4.12.2.8 getX()	30
4.12.2.9 getY()	30
4.12.2.10 handler()	30
4.12.2.11 handlerByTime()	30
4.12.2.12 reset()	30
4.12.2.13 setSpeed()	31
4.12.2.14 updateLineReading()	31
4.12.3 Member Data Documentation	31
4.12.3.1 angle	31
4.12.3.2 calibrationFinished	31
4.12.3.3 deltaDistance	31
4.12.3.4 distance	31
4.12.3.5 filteredAngle	31
4.12.3.6 lastDistance	32
4.12.3.7 lastLineSensorReading	32
4.12.3.8 lineSensor	32
4.12.3.9 lineSensorReading	32
4.12.3.10 motorD	32
4.12.3.11 motorE	32
4.12.3.12 ptRobot	32

4.12.3.13 xPos	32
4.12.3.14 yPos	33
4.13 Line_Sensor Class Reference	33
4.13.1 Constructor & Destructor Documentation	33
4.13.1.1 Line_Sensor() [1/2]	33
4.13.1.2 Line_Sensor() [2/2]	34
4.13.2 Member Function Documentation	34
4.13.2.1 calibrate()	34
4.13.2.2 read()	34
4.13.3 Member Data Documentation	34
4.13.3.1 Sensors	34
4.14 Motor Class Reference	34
4.14.1 Constructor & Destructor Documentation	35
4.14.1.1 Motor() [1/2]	35
4.14.1.2 Motor() [2/2]	36
4.14.2 Member Function Documentation	36
4.14.2.1 Get_Speed()	36
4.14.2.2 getDeltaDistance()	36
4.14.2.3 getDistance()	36
4.14.2.4 getE()	36
4.14.2.5 getSpeedRadS()	36
4.14.2.6 getTeta()	36
4.14.2.7 getU()	37
4.14.2.8 Handler()	37
4.14.2.9 Motor_Handler_by_time()	37
4.14.2.10 Motor_Initialiize()	37
4.14.2.11 reset()	37
4.14.2.12 Set_Speed()	37
4.14.3 Member Data Documentation	37
4.14.3.1 E	37
4.14.3.2 encoder	38
4.14.3.3 IN1	38
4.14.3.4 IN2	38
4.14.3.5 Motor_number	38
4.14.3.6 motPtr	38
4.14.3.7 pwmMotor	38
4.14.3.8 Speed_Reference	38
4.14.3.9 U	39
4.15 NVIC_Type Struct Reference	39
4.15.1 Member Data Documentation	39
4.15.1.1 IABR	39
4.15.1.2 ICER	39

4.15.1.3 ICPR	40
4.15.1.4 IP	40
4.15.1.5 ISER	40
4.15.1.6 ISPR	40
4.15.1.7 RESERVED0	40
4.15.1.8 RESERVED2	40
4.15.1.9 RESERVED3	40
4.15.1.10 RESERVED4	40
4.15.1.11 RESERVED5	41
4.15.1.12 RESERVED1	41
4.15.1.13 STIR	41
4.16 PWM Class Reference	41
4.16.1 Constructor & Destructor Documentation	41
4.16.1.1 PWM() [1/3]	42
4.16.1.2 PWM() [2/3]	42
4.16.1.3 PWM() [3/3]	42
4.16.2 Member Function Documentation	42
4.16.2.1 getWritedPWM()	42
4.16.2.2 PWMWrite()	42
4.17 PWR_TypeDef Struct Reference	42
4.17.1 Member Data Documentation	43
4.17.1.1 CR	43
4.17.1.2 CSR	43
4.18 RCC_TypeDef Struct Reference	43
4.18.1 Member Data Documentation	43
4.18.1.1 AHBENR	44
4.18.1.2 APB1ENR	44
4.18.1.3 APB1RSTR	44
4.18.1.4 APB2ENR	44
4.18.1.5 APB2RSTR	44
4.18.1.6 BDCR	44
4.18.1.7 CFGR	44
4.18.1.8 CIR	44
4.18.1.9 CR	45
4.18.1.10 CSR	45
4.19 Reflectance_Sensor Class Reference	45
4.19.1 Constructor & Destructor Documentation	45
4.19.1.1 Reflectance_Sensor() [1/3]	45
4.19.1.2 Reflectance_Sensor() [2/3]	46
4.19.1.3 Reflectance_Sensor() [3/3]	46
4.19.2 Member Function Documentation	46
4.19.2.1 Calib_Reflectance_Sensor()	46



4.19.2.2 Reflectance_Read()	46
4.19.3 Friends And Related Function Documentation	46
4.19.3.1 Line_Sensor	46
4.20 RTC_TypeDef Struct Reference	46
4.20.1 Member Data Documentation	47
4.20.1.1 ALRH	47
4.20.1.2 ALRL	47
4.20.1.3 CNTH	47
4.20.1.4 CNTL	47
4.20.1.5 CRH	48
4.20.1.6 CRL	48
4.20.1.7 DIVH	48
4.20.1.8 DIVL	48
4.20.1.9 PRLH	48
4.20.1.10 PRL	48
4.20.1.11 RESERVED0	48
4.20.1.12 RESERVED1	48
4.20.1.13 RESERVED2	49
4.20.1.14 RESERVED3	49
4.20.1.15 RESERVED4	49
4.20.1.16 RESERVED5	49
4.20.1.17 RESERVED6	49
4.20.1.18 RESERVED7	49
4.20.1.19 RESERVED8	49
4.20.1.20 RESERVED9	50
4.21 SysClock Class Reference	50
4.21.1 Constructor & Destructor Documentation	50
4.21.1.1 SysClock()	50
4.21.2 Member Function Documentation	50
4.21.2.1 MCO()	50
4.21.2.2 SysClockInit()	51
4.21.2.3 SysTickGetEvent()	51
4.21.2.4 SysTickInit()	51
4.22 SysTick_Type Struct Reference	52
4.22.1 Member Data Documentation	52
4.22.1.1 CALIB	52
4.22.1.2 CTRL	52
4.22.1.3 LOAD	52
4.22.1.4 VAL	52
4.23 TIM_TypeDef Struct Reference	53
4.23.1 Member Data Documentation	53
4.23.1.1 ARR	54

4.23.1.2 BDTR	54
4.23.1.3 CCER	54
4.23.1.4 CCMR1	54
4.23.1.5 CCMR2	54
4.23.1.6 CCR1	54
4.23.1.7 CCR2	54
4.23.1.8 CCR3	54
4.23.1.9 CCR4	55
4.23.1.10 CNT	55
4.23.1.11 CR1	55
4.23.1.12 CR2	55
4.23.1.13 DCR	55
4.23.1.14 DIER	55
4.23.1.15 DMAR	55
4.23.1.16 EGR	55
4.23.1.17 PSC	56
4.23.1.18 RCR	56
4.23.1.19 RESERVED0	56
4.23.1.20 RESERVED1	56
4.23.1.21 RESERVED10	56
4.23.1.22 RESERVED11	56
4.23.1.23 RESERVED12	56
4.23.1.24 RESERVED13	56
4.23.1.25 RESERVED14	57
4.23.1.26 RESERVED15	57
4.23.1.27 RESERVED16	57
4.23.1.28 RESERVED17	57
4.23.1.29 RESERVED18	57
4.23.1.30 RESERVED19	57
4.23.1.31 RESERVED2	57
4.23.1.32 RESERVED3	57
4.23.1.33 RESERVED4	58
4.23.1.34 RESERVED5	58
4.23.1.35 RESERVED6	58
4.23.1.36 RESERVED7	58
4.23.1.37 RESERVED8	58
4.23.1.38 RESERVED9	58
4.23.1.39 SMCR	58
4.23.1.40 SR	59
4.24 Timer Class Reference	59
4.24.1 Constructor & Destructor Documentation	60
4.24.1.1 Timer() [1/3]	60

4.24.1.2 Timer() [2/3]	60
4.24.1.3 Timer() [3/3]	60
4.24.2 Member Function Documentation	60
4.24.2.1 ClearInterruptFlag()	60
4.24.2.2 delay()	60
4.24.2.3 GetTim()	61
4.24.2.4 GetTIMChannel()	61
4.24.2.5 GetTime_milisec()	61
4.24.2.6 GetTime_usec()	61
4.24.2.7 GetTIMMode()	61
4.24.2.8 GetTIMRemap()	61
4.24.2.9 InterruptTime()	61
4.24.2.10 SetTim()	62
4.24.2.11 SetTIMChannel()	62
4.24.2.12 SetTIMMode()	62
4.24.2.13 SetTIMRemap()	62
4.24.2.14 Timer_Handler()	62
4.24.2.15 Timer_Initiallize()	62
4.24.2.16 TimerInit()	63
4.24.2.17 verifyTimeInterrupt()	63
4.24.3 Member Data Documentation	64
4.24.3.1 TIM	64
4.25 USART Class Reference	64
4.25.1 Constructor & Destructor Documentation	64
4.25.1.1 USART()	64
4.25.2 Member Function Documentation	64
4.25.2.1 Available()	65
4.25.2.2 Receive()	65
4.25.2.3 Send()	65
4.25.2.4 Send_Vec_16()	65
4.25.2.5 sendFloat()	65
4.25.2.6 sendUint16()	65
4.25.2.7 sendUint32()	65
4.26 USART_TypeDef Struct Reference	66
4.26.1 Detailed Description	66
4.26.2 Member Data Documentation	66
4.26.2.1 BRR	66
4.26.2.2 CR1	66
4.26.2.3 CR2	66
4.26.2.4 CR3	67
4.26.2.5 DR	67
4.26.2.6 GTPR	67

4.26.2.7 RESERVED0	67
4.26.2.8 RESERVED1	67
4.26.2.9 RESERVED2	67
4.26.2.10 RESERVED3	67
4.26.2.11 RESERVED4	67
4.26.2.12 RESERVED5	68
4.26.2.13 RESERVED6	68
4.26.2.14 SR	68
<b>5 File Documentation</b>	<b>69</b>
5.1 C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/ADC.cpp File Reference	69
5.2 C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/ADC.h File Reference	69
5.2.1 Enumeration Type Documentation	69
5.2.1.1 ADC_CHANNELS	69
5.2.1.2 ADC_CONVERSION_MODES	70
5.3 C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/ANGLE_LINE_SENSOR.cpp File Reference	70
5.4 C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/ANGLE_LINE_SENSOR.h File Reference	70
5.5 C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/ENCODER.cpp File Reference	70
5.6 C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/ENCODER.h File Reference	71
5.6.1 Macro Definition Documentation	71
5.6.1.1 AutoReload_Ticks	71
5.6.1.2 encoderFilterOrder	71
5.6.1.3 Max_delay_Ticks_Time	72
5.6.1.4 pi	72
5.6.1.5 pulsePerRevolution	72
5.6.1.6 Ticks_till_int	72
5.6.1.7 ticksToRad	72
5.6.2 Enumeration Type Documentation	72
5.6.2.1 ENCODER_DIRECTION	72
5.6.2.2 ENCODER_ENUM	73
5.7 C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/FOLLOWING_CONTROL↔ER.cpp File Reference	73
5.8 C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/FOLLOWING_CONTROL↔ER.h File Reference	73
5.8.1 Macro Definition Documentation	74
5.8.1.1 K1	74
5.8.1.2 K1a	74
5.8.1.3 K2	74
5.8.1.4 K2a	74
5.8.1.5 K3	74
5.8.1.6 K4	74

5.8.1.7 K4k . . . . .	74
5.8.1.8 Kteta . . . . .	75
5.8.1.9 Ktetak . . . . .	75
5.8.1.10 Kx . . . . .	75
5.8.1.11 Kxk . . . . .	75
5.8.1.12 Ky . . . . .	75
5.8.1.13 Kyk . . . . .	75
5.8.1.14 v_ref_ini . . . . .	75
5.8.1.15 w_ref_ini . . . . .	75
5.9 C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/GPIO.cpp File Reference . . .	76
5.10 C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/GPIO.h File Reference . . .	76
5.10.1 Macro Definition Documentation . . . . .	77
5.10.1.1 HIGH . . . . .	77
5.10.1.2 LOW . . . . .	77
5.10.2 Enumeration Type Documentation . . . . .	77
5.10.2.1 GPIO_IO_ENUM . . . . .	77
5.10.2.2 GPIO_MODES . . . . .	78
5.10.2.3 PIN_NUMBERS . . . . .	78
5.10.2.4 PU_PD_ENUM . . . . .	79
5.11 C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/KINEMATIC_CONTROL.cpp File Reference . . . . .	79
5.12 C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/KINEMATIC_CONTROL.h File Reference . . . . .	79
5.12.1 Macro Definition Documentation . . . . .	80
5.12.1.1 angleCorrection . . . . .	80
5.12.1.2 angleFilterOrder . . . . .	80
5.12.1.3 distanceCorrection . . . . .	80
5.12.1.4 integrationTime . . . . .	80
5.12.1.5 KDw . . . . .	80
5.12.1.6 KPw . . . . .	81
5.12.1.7 Kw . . . . .	81
5.12.1.8 L . . . . .	81
5.12.1.9 Pw . . . . .	81
5.12.1.10 ts_w . . . . .	81
5.13 C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/line_follower.cpp File Reference . . . . .	81
5.14 C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/LINE_SENSOR.cpp File Reference . . . . .	81
5.15 C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/LINE_SENSOR.h File Reference . . . . .	82
5.15.1 Macro Definition Documentation . . . . .	82
5.15.1.1 calib_sensores . . . . .	82
5.15.1.2 maxDistance . . . . .	82

5.15.1.3 sensorFilterOrder . . . . .	82
5.16 C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/main.cpp File Reference . .	83
5.17 C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/malha_aberta.cpp File Reference . . . . .	83
5.18 C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/malha_fechada.cpp File Reference . . . . .	83
5.18.1 Macro Definition Documentation . . . . .	84
5.18.1.1 steps . . . . .	84
5.18.1.2 stepTime . . . . .	84
5.18.1.3 testTime . . . . .	84
5.18.2 Function Documentation . . . . .	84
5.18.2.1 main() . . . . .	85
5.18.2.2 TIM1_UP_IRQHandler() . . . . .	85
5.18.2.3 TIM2_IRQHandler() . . . . .	85
5.18.2.4 TIM3_IRQHandler() . . . . .	85
5.18.2.5 TIM4_IRQHandler() . . . . .	85
5.18.3 Variable Documentation . . . . .	85
5.18.3.1 counter . . . . .	85
5.18.3.2 delta_time . . . . .	86
5.18.3.3 lineangle . . . . .	86
5.18.3.4 linevalue . . . . .	86
5.18.3.5 pwmtest . . . . .	86
5.18.3.6 RPM_value . . . . .	86
5.18.3.7 t_ini . . . . .	86
5.18.3.8 time . . . . .	86
5.18.3.9 time_test . . . . .	86
5.18.3.10 vtest . . . . .	87
5.18.3.11 wtest . . . . .	87
5.18.3.12 xtest . . . . .	87
5.18.3.13 ytest . . . . .	87
5.19 C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/Micro.c File Reference . . .	87
5.20 C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/Micro.h File Reference . . .	87
5.20.1 Macro Definition Documentation . . . . .	89
5.20.1.1 ADC1 . . . . .	89
5.20.1.2 ADC1_BASE . . . . .	90
5.20.1.3 ADC2 . . . . .	90
5.20.1.4 ADC2_BASE . . . . .	90
5.20.1.5 AFIO . . . . .	90
5.20.1.6 AFIO_BASE . . . . .	90
5.20.1.7 AHBPERIPH_BASE . . . . .	90
5.20.1.8 APB1PERIPH_BASE . . . . .	90
5.20.1.9 APB2PERIPH_BASE . . . . .	90
5.20.1.10 CoreDebug . . . . .	91

5.20.1.11 CoreDebug_BASE . . . . .	91
5.20.1.12 DWT . . . . .	91
5.20.1.13 DWT_BASE . . . . .	91
5.20.1.14 FLASH . . . . .	91
5.20.1.15 FLASH_BASE . . . . .	91
5.20.1.16 FLASH_R_BASE . . . . .	91
5.20.1.17 FSMC_R_BASE . . . . .	91
5.20.1.18 GPIOA . . . . .	92
5.20.1.19 GPIOA_BASE . . . . .	92
5.20.1.20 GPIOB . . . . .	92
5.20.1.21 GPIOB_BASE . . . . .	92
5.20.1.22 GPIOC . . . . .	92
5.20.1.23 GPIOC_BASE . . . . .	92
5.20.1.24 GPIOD . . . . .	92
5.20.1.25 GPIOD_BASE . . . . .	92
5.20.1.26 GPIOE . . . . .	93
5.20.1.27 GPIOE_BASE . . . . .	93
5.20.1.28 GPIOF . . . . .	93
5.20.1.29 GPIOF_BASE . . . . .	93
5.20.1.30 GPIOG . . . . .	93
5.20.1.31 GPIOG_BASE . . . . .	93
5.20.1.32 ITM . . . . .	93
5.20.1.33 ITM_BASE . . . . .	93
5.20.1.34 NVIC . . . . .	94
5.20.1.35 NVIC_BASE . . . . .	94
5.20.1.36 PERIPH_BASE . . . . .	94
5.20.1.37 PERIPH_BB_BASE . . . . .	94
5.20.1.38 PWR . . . . .	94
5.20.1.39 PWR_BASE . . . . .	94
5.20.1.40 RCC . . . . .	94
5.20.1.41 RCC_BASE . . . . .	94
5.20.1.42 RTC . . . . .	95
5.20.1.43 RTC_BASE . . . . .	95
5.20.1.44 SCB . . . . .	95
5.20.1.45 SCB_BASE . . . . .	95
5.20.1.46 SCnSCB . . . . .	95
5.20.1.47 SCS_BASE . . . . .	95
5.20.1.48 SRAM_BASE . . . . .	95
5.20.1.49 SRAM_BB_BASE . . . . .	95
5.20.1.50 SysTick . . . . .	96
5.20.1.51 SysTick_BASE . . . . .	96
5.20.1.52 TIM1 . . . . .	96

5.20.1.53 TIM1_BASE . . . . .	96
5.20.1.54 TIM2 . . . . .	96
5.20.1.55 TIM2_BASE . . . . .	96
5.20.1.56 TIM3 . . . . .	96
5.20.1.57 TIM3_BASE . . . . .	96
5.20.1.58 TIM4 . . . . .	97
5.20.1.59 TIM4_BASE . . . . .	97
5.20.1.60 TPI . . . . .	97
5.20.1.61 TPI_BASE . . . . .	97
5.20.1.62 USART1 . . . . .	97
5.20.1.63 USART1_BASE . . . . .	97
5.20.1.64 USART2 . . . . .	97
5.20.1.65 USART2_BASE . . . . .	97
5.20.1.66 USART3 . . . . .	98
5.20.1.67 USART3_BASE . . . . .	98
5.20.2 Typedef Documentation . . . . .	98
5.20.2.1 IRQn_Type . . . . .	98
5.20.3 Enumeration Type Documentation . . . . .	98
5.20.3.1 IRQn . . . . .	98
5.21 C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/MOTOR.cpp File Reference . . . . .	100
5.22 C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/MOTOR.h File Reference . . . . .	100
5.22.1 Macro Definition Documentation . . . . .	101
5.22.1.1 desired_size . . . . .	101
5.22.1.2 k . . . . .	101
5.22.1.3 K . . . . .	101
5.22.1.4 KI . . . . .	101
5.22.1.5 KP . . . . .	102
5.22.1.6 P . . . . .	102
5.22.1.7 r . . . . .	102
5.22.1.8 radsToRpm . . . . .	102
5.22.1.9 rpmToRads . . . . .	102
5.22.1.10 rpmToV . . . . .	102
5.22.1.11 ts . . . . .	102
5.22.1.12 Ts . . . . .	102
5.22.2 Enumeration Type Documentation . . . . .	102
5.22.2.1 MOTOR_ENUM . . . . .	102
5.23 C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/PWM.cpp File Reference . . . . .	103
5.24 C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/PWM.h File Reference . . . . .	103
5.24.1 Macro Definition Documentation . . . . .	103
5.24.1.1 AutoReloadPWM . . . . .	103
5.24.1.2 Max_PWM . . . . .	104
5.24.1.3 PWM_Frequency . . . . .	104



5.24.1.4 Timer_Frequency	104
5.25 C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/REFLECTANCE_SENS↔ OR.cpp File Reference	104
5.26 C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/REFLECTANCE_SENS↔ OR.h File Reference	104
5.27 C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/SERIAL_COMMUNICATI↔ ON.cpp File Reference	104
5.28 C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/SERIAL_COMMUNICATI↔ ON.h File Reference	105
5.28.1 Macro Definition Documentation	105
5.28.1.1 bitsToSend	105
5.28.1.2 robotRun	105
5.28.1.3 robotStop	106
5.28.1.4 samplingTime	106
5.28.1.5 sendEverything	106
5.28.1.6 sendingTime	106
5.28.1.7 sendLineReading	106
5.28.1.8 sendMotorDdata	106
5.28.1.9 sendMotorsSpeed	106
5.28.1.10 sendMotorsSpeedAndControl	106
5.28.1.11 sendMotorsSpeedControlAndErrors	107
5.28.1.12 sendPositionAndOrientation	107
5.28.1.13 sendPWMandEncoderDataRight	107
5.28.1.14 sendV_W	107
5.28.1.15 sendV_W_and_V_Wcontrol	107
5.28.1.16 sendV_Wcontrol	107
5.29 C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/SysClock.cpp File Reference	107
5.30 C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/SysClock.h File Reference	108
5.30.1 Enumeration Type Documentation	108
5.30.1.1 SysTickBaseTimeEnum	108
5.31 C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/testes.cpp File Reference	109
5.32 C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/TIMER.cpp File Reference	109
5.33 C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/TIMER.h File Reference	109
5.33.1 Macro Definition Documentation	109
5.33.1.1 AutoReload_Counter	109
5.33.1.2 Prescale	110
5.33.1.3 Time_between_int	110
5.33.1.4 Time_between_int_milis	110
5.33.2 Enumeration Type Documentation	110
5.33.2.1 TIM_CHANNELS	110
5.33.2.2 TIM_MODE	110
5.33.2.3 TIM_REMAP	111
5.34 C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/USART.cpp File Reference	111

5.34.1 Function Documentation . . . . .	111
5.34.1.1 sendUInt32() . . . . .	111
5.35 C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/USART.h File Reference . .	111
5.35.1 Enumeration Type Documentation . . . . .	112
5.35.1.1 BD_ENUM . . . . .	112
<b>Index</b>	<b>113</b>

# Chapter 1

## Hierarchical Index

### 1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

ADC_TypeDef . . . . .	8
AFIO_TypeDef . . . . .	12
Communication . . . . .	14
Controller . . . . .	15
FLASH_TypeDef . . . . .	20
GPIO . . . . .	22
ADC . . . . .	7
GPIO_STRUCT . . . . .	25
GPIO_TypeDef . . . . .	26
Kinematic . . . . .	27
Line_Sensor . . . . .	33
ANGLE_LINE_SENSOR_H . . . . .	13
Motor . . . . .	34
NVIC_Type . . . . .	39
PWR_TypeDef . . . . .	42
RCC_TypeDef . . . . .	43
Reflectance_Sensor . . . . .	45
RTC_TypeDef . . . . .	46
SysClock . . . . .	50
SysTick_Type . . . . .	52
TIM_TypeDef . . . . .	53
Timer . . . . .	59
Encoder . . . . .	17
PWM . . . . .	41
USART . . . . .	64
USART_TypeDef . . . . .	66



## Chapter 2

# Class Index

### 2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

ADC	7
ADC_TypeDef	8
AFIO_TypeDef	12
ANGLE_LINE_SENSOR_H	13
Communication	14
Controller	15
Encoder	17
FLASH_TypeDef	20
GPIO	22
GPIO_STRUCT	25
GPIO_TypeDef	26
Kinematic	27
Line_Sensor	33
Motor	34
NVIC_Type	39
PWM	41
PWR_TypeDef	42
RCC_TypeDef	43
Reflectance_Sensor	45
RTC_TypeDef	46
SysClock	50
SysTick_Type	52
TIM_TypeDef	53
Timer	59
USART	64
USART_TypeDef	
Universal Synchronous Asynchronous Receiver Transmitter	66



## Chapter 3

# File Index

### 3.1 File List

Here is a list of all files with brief descriptions:

C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/ADC.cpp	69
C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/ADC.h	69
C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/ANGLE_LINE_SENSOR.cpp	70
C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/ANGLE_LINE_SENSOR.h	70
C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/ENCODER.cpp	70
C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/ENCODER.h	71
C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/FOLLOWING_CONTROLLER.cpp	73
C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/FOLLOWING_CONTROLLER.h	73
C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/GPIO.cpp	76
C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/GPIO.h	76
C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/KINEMATIC_CONTROL.cpp	79
C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/KINEMATIC_CONTROL.h	79
C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/line_follower.cpp	81
C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/LINE_SENSOR.cpp	81
C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/LINE_SENSOR.h	82
C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/main.cpp	83
C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/malha_aberta.cpp	83
C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/malha_fechada.cpp	83
C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/Micro.c	87
C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/Micro.h	87
C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/MOTOR.cpp	100
C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/MOTOR.h	100
C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/PWM.cpp	103
C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/PWM.h	103
C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/REFLECTANCE_SENSOR.cpp	104
C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/REFLECTANCE_SENSOR.h	104
C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/SERIAL_COMMUNICATION.cpp	104
C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/SERIAL_COMMUNICATION.h	105
C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/SysClock.cpp	107
C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/SysClock.h	108
C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/testes.cpp	109
C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/TIMER.cpp	109
C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/TIMER.h	109
C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/USART.cpp	111
C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/USART.h	111





## Chapter 4

# Class Documentation

### 4.1 ADC Class Reference

```
#include <ADC.h>
```

Inheritance diagram for ADC:



#### Public Member Functions

- [ADC](#) ([ADC\\_CHANNELS](#) ADCChannel)
- `uint16_t` [analogRead](#) ()
- [ADC\\_CHANNELS](#) [GetADCChannel](#) ()

#### Protected Member Functions

- `void` [ConfigADCPin](#) ()
- `void` [SetADCChannel](#) ([ADC\\_CHANNELS](#) ADCChannel)

#### 4.1.1 Constructor & Destructor Documentation

##### 4.1.1.1 ADC()

```
ADC::ADC (
    ADC\_CHANNELS ADCChannel )
```

## 4.1.2 Member Function Documentation

### 4.1.2.1 analogRead()

```
uint16_t ADC::analogRead ( )
```

< Peripheral base address in the alias region

< Peripheral base address in the alias region

< Peripheral base address in the alias region

< Peripheral base address in the alias region

< Peripheral base address in the alias region

### 4.1.2.2 ConfigADCPin()

```
void ADC::ConfigADCPin ( ) [protected]
```

< Peripheral base address in the alias region

< Peripheral base address in the alias region

< Peripheral base address in the alias region

### 4.1.2.3 GetADCChannel()

```
ADC\_CHANNELS ADC::GetADCChannel ( )
```

### 4.1.2.4 SetADCChannel()

```
void ADC::SetADCChannel (
    ADC\_CHANNELS ADCChannel ) [protected]
```

The documentation for this class was generated from the following files:

- C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/[ADC.h](#)
- C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/[ADC.cpp](#)

## 4.2 ADC\_TypeDef Struct Reference

```
#include <Micro.h>
```

## Public Attributes

- volatile uint32\_t [SR](#)
- volatile uint32\_t [CR1](#)
- volatile uint32\_t [CR2](#)
- volatile uint32\_t [SMPR1](#)
- volatile uint32\_t [SMPR2](#)
- volatile uint32\_t [JOFR1](#)
- volatile uint32\_t [JOFR2](#)
- volatile uint32\_t [JOFR3](#)
- volatile uint32\_t [JOFR4](#)
- volatile uint32\_t [HTR](#)
- volatile uint32\_t [LTR](#)
- volatile uint32\_t [SQR1](#)
- volatile uint32\_t [SQR2](#)
- volatile uint32\_t [SQR3](#)
- volatile uint32\_t [JSQR](#)
- volatile uint32\_t [JDR1](#)
- volatile uint32\_t [JDR2](#)
- volatile uint32\_t [JDR3](#)
- volatile uint32\_t [JDR4](#)
- volatile uint32\_t [DR](#)

## 4.2.1 Member Data Documentation

### 4.2.1.1 CR1

```
volatile uint32_t ADC_TypeDef::CR1
```

### 4.2.1.2 CR2

```
volatile uint32_t ADC_TypeDef::CR2
```

### 4.2.1.3 DR

```
volatile uint32_t ADC_TypeDef::DR
```

### 4.2.1.4 HTR

```
volatile uint32_t ADC_TypeDef::HTR
```

#### 4.2.1.5 JDR1

```
volatile uint32_t ADC_TypeDef::JDR1
```

#### 4.2.1.6 JDR2

```
volatile uint32_t ADC_TypeDef::JDR2
```

#### 4.2.1.7 JDR3

```
volatile uint32_t ADC_TypeDef::JDR3
```

#### 4.2.1.8 JDR4

```
volatile uint32_t ADC_TypeDef::JDR4
```

#### 4.2.1.9 JOFR1

```
volatile uint32_t ADC_TypeDef::JOFR1
```

#### 4.2.1.10 JOFR2

```
volatile uint32_t ADC_TypeDef::JOFR2
```

#### 4.2.1.11 JOFR3

```
volatile uint32_t ADC_TypeDef::JOFR3
```

#### 4.2.1.12 JOFR4

```
volatile uint32_t ADC_TypeDef::JOFR4
```

#### 4.2.1.13 JSQR

```
volatile uint32_t ADC_TypeDef::JSQR
```

#### 4.2.1.14 LTR

```
volatile uint32_t ADC_TypeDef::LTR
```

#### 4.2.1.15 SMPR1

```
volatile uint32_t ADC_TypeDef::SMPR1
```

#### 4.2.1.16 SMPR2

```
volatile uint32_t ADC_TypeDef::SMPR2
```

#### 4.2.1.17 SQR1

```
volatile uint32_t ADC_TypeDef::SQR1
```

#### 4.2.1.18 SQR2

```
volatile uint32_t ADC_TypeDef::SQR2
```

#### 4.2.1.19 SQR3

```
volatile uint32_t ADC_TypeDef::SQR3
```

#### 4.2.1.20 SR

```
volatile uint32_t ADC_TypeDef::SR
```

The documentation for this struct was generated from the following file:

- C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/[Micro.h](#)

### 4.3 AFIO\_TypeDef Struct Reference

```
#include <Micro.h>
```

#### Public Attributes

- volatile uint32\_t [EVCR](#)
- volatile uint32\_t [MAPR](#)
- volatile uint32\_t [EXTICR](#) [4]
- uint32\_t [RESERVED0](#)
- volatile uint32\_t [MAPR2](#)

#### 4.3.1 Member Data Documentation

##### 4.3.1.1 EVCR

```
volatile uint32_t AFIO_TypeDef::EVCR
```

##### 4.3.1.2 EXTICR

```
volatile uint32_t AFIO_TypeDef::EXTICR[4]
```

##### 4.3.1.3 MAPR

```
volatile uint32_t AFIO_TypeDef::MAPR
```

#### 4.3.1.4 MAPR2

```
volatile uint32_t AFIO_TypeDef::MAPR2
```

#### 4.3.1.5 RESERVED0

```
uint32_t AFIO_TypeDef::RESERVED0
```

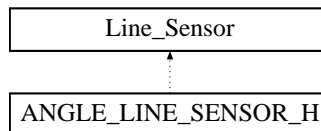
The documentation for this struct was generated from the following file:

- C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/[Micro.h](#)

## 4.4 ANGLE\_LINE\_SENSOR\_H Class Reference

```
#include <ANGLE_LINE_SENSOR.h>
```

Inheritance diagram for ANGLE\_LINE\_SENSOR\_H:



### Public Member Functions

- float [readAngle](#) ()
- void [updateLineAngle](#) ()

### 4.4.1 Member Function Documentation

#### 4.4.1.1 readAngle()

```
float ANGLE_LINE_SENSOR_H::readAngle ( )
```

#### 4.4.1.2 updateLineAngle()

```
void ANGLE_LINE_SENSOR_H::updateLineAngle ( )
```

The documentation for this class was generated from the following file:

- C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/[ANGLE\\_LINE\\_SENSOR.h](#)

## 4.5 Communication Class Reference

```
#include <SERIAL_COMMUNICATION.h>
```

### Public Member Functions

- [Communication](#) ([Controller](#) \*LineFollower\_, [USART](#) \*Serial\_)
- void [waitForCommand](#) ()
- void [verifyCommand](#) ()

### Static Public Member Functions

- static void [HandlerByTime](#) ()

### Public Attributes

- uint8\_t [command](#) { 0 }

## 4.5.1 Constructor & Destructor Documentation

### 4.5.1.1 Communication()

```
Communication::Communication (
    Controller * LineFollower_,
    USART * Serial_ )
```

## 4.5.2 Member Function Documentation

### 4.5.2.1 HandlerByTime()

```
void Communication::HandlerByTime ( ) [static]
```

### 4.5.2.2 verifyCommand()

```
void Communication::verifyCommand ( )
```



#### 4.5.2.3 waitForCommand()

```
void Communication::waitForCommand ( )
```

### 4.5.3 Member Data Documentation

#### 4.5.3.1 command

```
uint8_t Communication::command { 0 }
```

The documentation for this class was generated from the following files:

- C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/[SERIAL\\_COMMUNICATION.h](#)
- C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/[SERIAL\\_COMMUNICATION.cpp](#)

## 4.6 Controller Class Reference

```
#include <FOLLOWING_CONTROLLER.h>
```

### Public Member Functions

- [Controller](#) (Kinematic Robot)
- void [setSpeedRef](#) (float vr, float wr)
- void [controlRule](#) ()
- void [reset](#) ()
- float [getVcontrol](#) ()
- float [getWcontrol](#) ()
- void [start](#) (float vr, float wr)
- void [stop](#) ()

### Static Public Member Functions

- static void [HandlerByTime](#) ()

### Public Attributes

- [Kinematic Robot](#)

#### 4.6.1 Constructor & Destructor Documentation

#### 4.6.1.1 Controller()

```
Controller::Controller (
    Kinematic Robot )
```

### 4.6.2 Member Function Documentation

#### 4.6.2.1 controlRule()

```
void Controller::controlRule ( )
```

#### 4.6.2.2 getVcontrol()

```
float Controller::getVcontrol ( )
```

#### 4.6.2.3 getWcontrol()

```
float Controller::getWcontrol ( )
```

#### 4.6.2.4 HandlerByTime()

```
void Controller::HandlerByTime ( ) [static]
```

#### 4.6.2.5 reset()

```
void Controller::reset ( )
```

#### 4.6.2.6 setSpeedRef()

```
void Controller::setSpeedRef (
    float vr,
    float wr )
```

#### 4.6.2.7 start()

```
void Controller::start (
    float vr,
    float wr )
```

#### 4.6.2.8 stop()

```
void Controller::stop ( )
```

### 4.6.3 Member Data Documentation

#### 4.6.3.1 Robot

`Kinematic` Controller::Robot

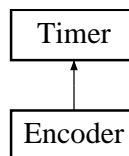
The documentation for this class was generated from the following files:

- C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/[FOLLOWING\\_CONTROLLER.h](#)
- C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/[FOLLOWING\\_CONTROLLER.cpp](#)

## 4.7 Encoder Class Reference

```
#include <ENCODER.h>
```

Inheritance diagram for Encoder:



### Public Member Functions

- [Encoder](#) ([TIM\\_TypeDef](#) \*TIM)
- [Encoder](#) ([Encoder](#) \*encoder)
- void [reset](#) ()
- float [getTicks](#) ()
- float [getDeltaTicks](#) ()
- [uint32\\_t](#) [getTicksTime](#) ()
- [uint32\\_t](#) [getLastTicksTime](#) ()
- [uint32\\_t](#) [getDeltaTime](#) ()
- float [getSpeed](#) ()
- float [getNotFilteredSpeed](#) ()
- float [getLastSpeed](#) ()
- float [getTeta](#) ()
- bool [getDirection](#) ()

## Static Public Member Functions

- static void [Encoder\\_Initialize](#) ()
- static void [Encoder\\_Handler](#) ([ENCODER\\_ENUM](#) enc)
- static void [Encoder\\_Handler\\_by\\_Time](#) ()

## Additional Inherited Members

### 4.7.1 Constructor & Destructor Documentation

#### 4.7.1.1 Encoder() [1/2]

```
Encoder::Encoder (
    TIM\_TypeDef * TIM )
```

#### 4.7.1.2 Encoder() [2/2]

```
Encoder::Encoder (
    Encoder * encoder )
```

### 4.7.2 Member Function Documentation

#### 4.7.2.1 Encoder\_Handler()

```
void Encoder::Encoder_Handler (
    ENCODER\_ENUM enc ) [static]
```

#### 4.7.2.2 Encoder\_Handler\_by\_Time()

```
void Encoder::Encoder_Handler_by_Time ( ) [static]
```

#### 4.7.2.3 Encoder\_Initialize()

```
void Encoder::Encoder_Initialize ( ) [static]
```

#### 4.7.2.4 getDeltaTicks()

```
float Encoder::getDeltaTicks ( )
```

#### 4.7.2.5 getDeltaTime()

```
uint32_t Encoder::getDeltaTime ( )
```

#### 4.7.2.6 getDirection()

```
bool Encoder::getDirection ( )
```

#### 4.7.2.7 getLastSpeed()

```
float Encoder::getLastSpeed ( )
```

#### 4.7.2.8 getLastTicksTime()

```
uint32_t Encoder::getLastTicksTime ( )
```

#### 4.7.2.9 getNotFilteredSpeed()

```
float Encoder::getNotFilteredSpeed ( )
```

#### 4.7.2.10 getSpeed()

```
float Encoder::getSpeed ( )
```

#### 4.7.2.11 getTeta()

```
float Encoder::getTeta ( )
```

#### 4.7.2.12 getTicks()

```
float Encoder::getTicks ( )
```

#### 4.7.2.13 getTicksTime()

```
uint32_t Encoder::getTicksTime ( )
```

#### 4.7.2.14 reset()

```
void Encoder::reset ( )
```

The documentation for this class was generated from the following files:

- C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/[ENCODER.h](#)
- C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/[ENCODER.cpp](#)

## 4.8 FLASH\_TypeDef Struct Reference

```
#include <Micro.h>
```

### Public Attributes

- volatile uint32\_t [ACR](#)
- volatile uint32\_t [KEYR](#)
- volatile uint32\_t [OPTKEYR](#)
- volatile uint32\_t [SR](#)
- volatile uint32\_t [CR](#)
- volatile uint32\_t [AR](#)
- volatile uint32\_t [RESERVED](#)
- volatile uint32\_t [OBR](#)
- volatile uint32\_t [WRPR](#)

### 4.8.1 Member Data Documentation

#### 4.8.1.1 ACR

```
volatile uint32_t FLASH_TypeDef::ACR
```

#### 4.8.1.2 AR

```
volatile uint32_t FLASH_TypeDef::AR
```

#### 4.8.1.3 CR

```
volatile uint32_t FLASH_TypeDef::CR
```

#### 4.8.1.4 KEYR

```
volatile uint32_t FLASH_TypeDef::KEYR
```

#### 4.8.1.5 OBR

```
volatile uint32_t FLASH_TypeDef::OBR
```

#### 4.8.1.6 OPTKEYR

```
volatile uint32_t FLASH_TypeDef::OPTKEYR
```

#### 4.8.1.7 RESERVED

```
volatile uint32_t FLASH_TypeDef::RESERVED
```

#### 4.8.1.8 SR

```
volatile uint32_t FLASH_TypeDef::SR
```

#### 4.8.1.9 WRPR

```
volatile uint32_t FLASH_TypeDef::WRPR
```

The documentation for this struct was generated from the following file:

- C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/[Micro.h](#)

## 4.9 GPIO Class Reference

```
#include <GPIO.h>
```

Inheritance diagram for GPIO:



### Public Member Functions

- [GPIO \(\)](#)
- [GPIO \(GPIO\\_IO\\_ENUM IO\\_Pin, GPIO\\_MODES GPIOMode\)](#)
- [GPIO \(GPIO \\*gpio\)](#)
- void [SetGPIOPortPin \(GPIO\\_IO\\_ENUM IO\\_Pin\)](#)
- void [SetGPIOMode \(GPIO\\_MODES GPIOMode\)](#)
- void [SetIOPin \(GPIO\\_IO\\_ENUM IO\\_Pin\)](#)
- [GPIO\\_TypeDef \\* GetGPIOPort \(\)](#)
- [PIN\\_NUMBERS GetGPIOPinNumber \(\)](#)
- [GPIO\\_MODES GetGPIOMode \(\)](#)
- [PU\\_PD\\_ENUM GetGPIOPuPd \(\)](#)
- [GPIO\\_IO\\_ENUM GetIOPin \(\)](#)
- bool [GetGPIOState \(\)](#)
- void [ConfigGPIOPin \(\)](#)
- void [Config\\_PU\\_PD \(PU\\_PD\\_ENUM PU\\_PD\)](#)
- void [digitalWrite \(bool GPIOState\)](#)
- void [togglePin \(\)](#)
- bool [digitalRead \(\)](#)

### 4.9.1 Constructor & Destructor Documentation

#### 4.9.1.1 GPIO() [1/3]

```
GPIO::GPIO ( ) [inline]
```



#### 4.9.1.2 GPIO() [2/3]

```
GPIO::GPIO (
    GPIO_IO_ENUM IO_Pin,
    GPIO_MODES GPIOMode )
```

#### 4.9.1.3 GPIO() [3/3]

```
GPIO::GPIO (
    GPIO * gpio )
```

### 4.9.2 Member Function Documentation

#### 4.9.2.1 Config\_PU\_PD()

```
void GPIO::Config_PU_PD (
    PU_PD_ENUM PU_PD )
```

#### 4.9.2.2 ConfigGPIOPin()

```
void GPIO::ConfigGPIOPin ( )
```

< Peripheral base address in the alias region

< Peripheral base address in the alias region

< Peripheral base address in the alias region

< Peripheral base address in the alias region

< Peripheral base address in the alias region

< Peripheral base address in the alias region

< Peripheral base address in the alias region

< Peripheral base address in the alias region

< Peripheral base address in the alias region

< Peripheral base address in the alias region

< Peripheral base address in the alias region

< Peripheral base address in the alias region

< Peripheral base address in the alias region

< Peripheral base address in the alias region

< Peripheral base address in the alias region

< Peripheral base address in the alias region

< Peripheral base address in the alias region

< Peripheral base address in the alias region

< Peripheral base address in the alias region

< Peripheral base address in the alias region

#### 4.9.2.3 digitalRead()

```
bool GPIO::digitalRead ( )
```

#### 4.9.2.4 digitalWrite()

```
void GPIO::digitalWrite (
    bool GPIOState )
```

#### 4.9.2.5 GetGPIOMode()

```
GPIO\_MODES GPIO::GetGPIOMode ( )
```

#### 4.9.2.6 GetGPIOPinNumber()

```
PIN\_NUMBERS GPIO::GetGPIOPinNumber ( )
```

#### 4.9.2.7 GetGPIOPort()

```
GPIO\_TypeDef * GPIO::GetGPIOPort ( )
```

#### 4.9.2.8 GetGPIOPuPd()

```
PU\_PD\_ENUM GPIO::GetGPIOPuPd ( )
```

#### 4.9.2.9 GetGPIOState()

```
bool GPIO::GetGPIOState ( )
```

#### 4.9.2.10 GetIOPin()

```
GPIO_IO_ENUM GPIO::GetIOPin ( )
```

#### 4.9.2.11 SetGPIOMode()

```
void GPIO::SetGPIOMode (
    GPIO_MODES GPIOMode )
```

#### 4.9.2.12 SetGPIOPortPin()

```
void GPIO::SetGPIOPortPin (
    GPIO_IO_ENUM IO_Pin )
```

#### 4.9.2.13 SetIOPin()

```
void GPIO::SetIOPin (
    GPIO_IO_ENUM IO_Pin )
```

#### 4.9.2.14 togglePin()

```
void GPIO::togglePin ( )
```

The documentation for this class was generated from the following files:

- C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/[GPIO.h](#)
- C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/[GPIO.cpp](#)

## 4.10 GPIO\_STRUCT Struct Reference

```
#include <GPIO.h>
```

### Public Attributes

- [GPIO\\_TypeDef](#) \* [Port](#)
- [PIN\\_NUMBERS](#) [pinNumber](#)

## 4.10.1 Member Data Documentation

### 4.10.1.1 pinNumber

`PIN_NUMBERS GPIO_STRUCT::pinNumber`

### 4.10.1.2 Port

`GPIO_TypeDef* GPIO_STRUCT::Port`

The documentation for this struct was generated from the following file:

- `C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/GPIO.h`

## 4.11 GPIO\_TypeDef Struct Reference

```
#include <Micro.h>
```

### Public Attributes

- volatile uint32\_t `CRL`
- volatile uint32\_t `CRH`
- volatile uint32\_t `IDR`
- volatile uint32\_t `ODR`
- volatile uint32\_t `BSRR`
- volatile uint32\_t `BRR`
- volatile uint32\_t `LCKR`

## 4.11.1 Member Data Documentation

### 4.11.1.1 BRR

`volatile uint32_t GPIO_TypeDef::BRR`

#### 4.11.1.2 BSRR

```
volatile uint32_t GPIO_TypeDef::BSRR
```

#### 4.11.1.3 CRH

```
volatile uint32_t GPIO_TypeDef::CRH
```

#### 4.11.1.4 CRL

```
volatile uint32_t GPIO_TypeDef::CRL
```

#### 4.11.1.5 IDR

```
volatile uint32_t GPIO_TypeDef::IDR
```

#### 4.11.1.6 LCKR

```
volatile uint32_t GPIO_TypeDef::LCKR
```

#### 4.11.1.7 ODR

```
volatile uint32_t GPIO_TypeDef::ODR
```

The documentation for this struct was generated from the following file:

- C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/[Micro.h](#)

## 4.12 Kinematic Class Reference

```
#include <KINEMATIC_CONTROL.h>
```

## Public Member Functions

- void [handler](#) ()
- [Kinematic](#) ([Motor](#) motorD, [Motor](#) motorE, [Line\\_Sensor](#) lineSensor)
- [Kinematic](#) ([Kinematic](#) \*kinematic)
- void [reset](#) ()
- void [setSpeed](#) (float V, float w)
- float [getX](#) ()
- float [getY](#) ()
- float [getTeta](#) ()
- float [getLineAngle](#) ()
- float [getLinePosition](#) ()
- float [getLineAngleNotFiltered](#) ()
- float [getV](#) ()
- float [getW](#) ()
- void [calibrateLineSensor](#) (uint32\_t iterations)
- void [updateLineReading](#) ()

## Static Public Member Functions

- static void [handlerByTime](#) ()

## Public Attributes

- [Motor](#) motorD
- [Motor](#) motorE
- [Line\\_Sensor](#) lineSensor
- bool [calibrationFinished](#) {0}
- float [lineSensorReading](#) {0}
- float [lastLineSensorReading](#) {0}
- float [distance](#) {0}
- float [deltaDistance](#) {0}
- float [lastDistance](#) {0}
- float [xPos](#) {0}
- float [yPos](#) {0}
- float [angle](#) [2] {0}
- float [filteredAngle](#) {0}

## Static Public Attributes

- static [Kinematic](#) \* [ptRobot](#)

### 4.12.1 Constructor & Destructor Documentation

#### 4.12.1.1 Kinematic() [1/2]

```
Kinematic::Kinematic (
    Motor motorD,
    Motor motorE,
    Line_Sensor lineSensor )
```

#### 4.12.1.2 Kinematic() [2/2]

```
Kinematic::Kinematic (
    Kinematic * kinematic )
```

### 4.12.2 Member Function Documentation

#### 4.12.2.1 calibrateLineSensor()

```
void Kinematic::calibrateLineSensor (
    uint32_t iterations )
```

#### 4.12.2.2 getLineAngle()

```
float Kinematic::getLineAngle ( )
```

#### 4.12.2.3 getLineAngleNotFiltered()

```
float Kinematic::getLineAngleNotFiltered ( )
```

#### 4.12.2.4 getLinePosition()

```
float Kinematic::getLinePosition ( )
```

**4.12.2.5 getTeta()**

```
float Kinematic::getTeta ( )
```

**4.12.2.6 getV()**

```
float Kinematic::getV ( )
```

**4.12.2.7 getW()**

```
float Kinematic::getW ( )
```

**4.12.2.8 getX()**

```
float Kinematic::getX ( )
```

**4.12.2.9 getY()**

```
float Kinematic::getY ( )
```

**4.12.2.10 handler()**

```
void Kinematic::handler ( )
```

**4.12.2.11 handlerByTime()**

```
void Kinematic::handlerByTime ( ) [static]
```

**4.12.2.12 reset()**

```
void Kinematic::reset ( )
```



#### 4.12.2.13 setSpeed()

```
void Kinematic::setSpeed (
    float V,
    float w )
```

#### 4.12.2.14 updateLineReading()

```
void Kinematic::updateLineReading ( )
```

### 4.12.3 Member Data Documentation

#### 4.12.3.1 angle

```
float Kinematic::angle[2] {0}
```

#### 4.12.3.2 calibrationFinished

```
bool Kinematic::calibrationFinished {0}
```

#### 4.12.3.3 deltaDistance

```
float Kinematic::deltaDistance {0}
```

#### 4.12.3.4 distance

```
float Kinematic::distance {0}
```

#### 4.12.3.5 filteredAngle

```
float Kinematic::filteredAngle {0}
```

#### 4.12.3.6 lastDistance

```
float Kinematic::lastDistance {0}
```

#### 4.12.3.7 lastLineSensorReading

```
float Kinematic::lastLineSensorReading {0}
```

#### 4.12.3.8 lineSensor

```
Line_Sensor Kinematic::lineSensor
```

#### 4.12.3.9 lineSensorReading

```
float Kinematic::lineSensorReading {0}
```

#### 4.12.3.10 motorD

```
Motor Kinematic::motorD
```

#### 4.12.3.11 motorE

```
Motor Kinematic::motorE
```

#### 4.12.3.12 ptRobot

```
Kinematic * Kinematic::ptRobot [static]
```

#### 4.12.3.13 xPos

```
float Kinematic::xPos {0}
```

## 4.12.3.14 yPos

```
float Kinematic::yPos {0}
```

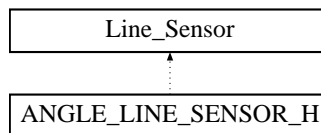
The documentation for this class was generated from the following files:

- C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/KINEMATIC\_CONTROL.h
- C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/KINEMATIC\_CONTROL.cpp

## 4.13 Line\_Sensor Class Reference

```
#include <LINE_SENSOR.h>
```

Inheritance diagram for Line\_Sensor:



## Public Member Functions

- [Line\\_Sensor](#) ([Reflectance\\_Sensor](#) Sensor1, [Reflectance\\_Sensor](#) Sensor2, [Reflectance\\_Sensor](#) Sensor3, [Reflectance\\_Sensor](#) Sensor4, [Reflectance\\_Sensor](#) Sensor5, [Reflectance\\_Sensor](#) Sensor6, [Reflectance\\_Sensor](#) Sensor7, [Reflectance\\_Sensor](#) Sensor8)
- [Line\\_Sensor](#) ([Line\\_Sensor](#) \*lineSensor)
- void [calibrate](#) (uint32\_t iterations)
- float [read](#) ()

## Public Attributes

- [Reflectance\\_Sensor](#) Sensors [8]

## 4.13.1 Constructor &amp; Destructor Documentation

## 4.13.1.1 Line\_Sensor() [1/2]

```
Line_Sensor::Line_Sensor (
    Reflectance\_Sensor Sensor1,
    Reflectance\_Sensor Sensor2,
    Reflectance\_Sensor Sensor3,
    Reflectance\_Sensor Sensor4,
    Reflectance\_Sensor Sensor5,
    Reflectance\_Sensor Sensor6,
    Reflectance\_Sensor Sensor7,
    Reflectance\_Sensor Sensor8 )
```

#### 4.13.1.2 Line\_Sensor() [2/2]

```
Line_Sensor::Line_Sensor (
    Line_Sensor * lineSensor )
```

### 4.13.2 Member Function Documentation

#### 4.13.2.1 calibrate()

```
void Line_Sensor::calibrate (
    uint32_t iterations )
```

#### 4.13.2.2 read()

```
float Line_Sensor::read ( )
```

### 4.13.3 Member Data Documentation

#### 4.13.3.1 Sensors

```
Reflectance_Sensor Line_Sensor::Sensors[8]
```

The documentation for this class was generated from the following files:

- C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/[LINE\\_SENSOR.h](#)
- C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/[LINE\\_SENSOR.cpp](#)

## 4.14 Motor Class Reference

```
#include <MOTOR.h>
```

## Public Member Functions

- void [Handler](#) ()
- [Motor](#) ([PWM pwmMotor](#), [Encoder encoder](#), [GPIO In1](#), [GPIO In2](#))
- [Motor](#) ([Motor \\*motor](#))
- void [reset](#) ()
- void [Set\\_Speed](#) (float [Speed\\_Reference](#))
- float [Get\\_Speed](#) ()
- float [getDistance](#) ()
- float [getDeltaDistance](#) ()
- float [getSpeedRadS](#) ()
- float [getU](#) ()
- float [getE](#) ()
- float [getTeta](#) ()

## Static Public Member Functions

- static void [Motor\\_Initialiize](#) ()
- static void [Motor\\_Handler\\_by\\_time](#) ()

## Public Attributes

- [PWM pwmMotor](#)
- [Encoder encoder](#)
- [GPIO IN1](#)
- [GPIO IN2](#)
- [MOTOR\\_ENUM Motor\\_number](#)
- float [U](#) [3] {0}
- float [E](#) [3] {0}
- float [Speed\\_Reference](#) {0}

## Static Public Attributes

- static [Motor \\* motPtr](#) [[Number\\_of\\_Motor](#)]

### 4.14.1 Constructor & Destructor Documentation

#### 4.14.1.1 [Motor\(\)](#) [1/2]

```
Motor::Motor (
    PWM pwmMotor,
    Encoder encoder,
    GPIO In1,
    GPIO In2 )
```

< Peripheral base address in the alias region

< Peripheral base address in the alias region

< Peripheral base address in the alias region

< Peripheral base address in the alias region

#### 4.14.1.2 Motor() [2/2]

```
Motor::Motor (
    Motor * motor )
```

### 4.14.2 Member Function Documentation

#### 4.14.2.1 Get\_Speed()

```
float Motor::Get_Speed ( )
```

#### 4.14.2.2 getDeltaDistance()

```
float Motor::getDeltaDistance ( )
```

#### 4.14.2.3 getDistance()

```
float Motor::getDistance ( )
```

#### 4.14.2.4 getE()

```
float Motor::getE ( )
```

#### 4.14.2.5 getSpeedRadS()

```
float Motor::getSpeedRadS ( )
```

#### 4.14.2.6 getTeta()

```
float Motor::getTeta ( )
```

#### 4.14.2.7 getU()

```
float Motor::getU ( )
```

#### 4.14.2.8 Handler()

```
void Motor::Handler ( )
```

#### 4.14.2.9 Motor\_Handler\_by\_time()

```
void Motor::Motor_Handler_by_time ( ) [static]
```

#### 4.14.2.10 Motor\_Initialiize()

```
void Motor::Motor_Initialiize ( ) [static]
```

#### 4.14.2.11 reset()

```
void Motor::reset ( )
```

#### 4.14.2.12 Set\_Speed()

```
void Motor::Set_Speed (
    float Speed_Reference )
```

### 4.14.3 Member Data Documentation

#### 4.14.3.1 E

```
float Motor::E[3] {0}
```

#### 4.14.3.2 encoder

`Encoder` `Motor::encoder`

#### 4.14.3.3 IN1

`GPIO` `Motor::IN1`

#### 4.14.3.4 IN2

`GPIO` `Motor::IN2`

#### 4.14.3.5 Motor\_number

`MOTOR_ENUM` `Motor::Motor_number`

#### 4.14.3.6 motPtr

`Motor` \* `Motor::motPtr` [static]

#### 4.14.3.7 pwmMotor

`PWM` `Motor::pwmMotor`

#### 4.14.3.8 Speed\_Reference

`float` `Motor::Speed_Reference` {0}



#### 4.14.3.9 U

```
float Motor::U[3] {0}
```

The documentation for this class was generated from the following files:

- C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/[MOTOR.h](#)
- C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/[MOTOR.cpp](#)

## 4.15 NVIC\_Type Struct Reference

```
#include <Micro.h>
```

### Public Attributes

- volatile uint32\_t [ISER](#) [8U]
- uint32\_t [RESERVED0](#) [24U]
- volatile uint32\_t [ICER](#) [8U]
- uint32\_t [RSERVED1](#) [24U]
- volatile uint32\_t [ISPR](#) [8U]
- uint32\_t [RESERVED2](#) [24U]
- volatile uint32\_t [ICPR](#) [8U]
- uint32\_t [RESERVED3](#) [24U]
- volatile uint32\_t [IABR](#) [8U]
- uint32\_t [RESERVED4](#) [56U]
- volatile uint8\_t [IP](#) [240U]
- uint32\_t [RESERVED5](#) [644U]
- volatile uint32\_t [STIR](#)

### 4.15.1 Member Data Documentation

#### 4.15.1.1 IABR

```
volatile uint32_t NVIC_Type::IABR[8U]
```

Offset: 0x200 (R/W) Interrupt Active bit Register

#### 4.15.1.2 ICER

```
volatile uint32_t NVIC_Type::ICER[8U]
```

Offset: 0x080 (R/W) Interrupt Clear Enable Register

#### 4.15.1.3 ICPR

```
volatile uint32_t NVIC_Type::ICPR[8U]
```

Offset: 0x180 (R/W) Interrupt Clear Pending Register

#### 4.15.1.4 IP

```
volatile uint8_t NVIC_Type::IP[240U]
```

Offset: 0x300 (R/W) Interrupt Priority Register (8Bit wide)

#### 4.15.1.5 ISER

```
volatile uint32_t NVIC_Type::ISER[8U]
```

Offset: 0x000 (R/W) Interrupt Set Enable Register

#### 4.15.1.6 ISPR

```
volatile uint32_t NVIC_Type::ISPR[8U]
```

Offset: 0x100 (R/W) Interrupt Set Pending Register

#### 4.15.1.7 RESERVED0

```
uint32_t NVIC_Type::RESERVED0[24U]
```

#### 4.15.1.8 RESERVED2

```
uint32_t NVIC_Type::RESERVED2[24U]
```

#### 4.15.1.9 RESERVED3

```
uint32_t NVIC_Type::RESERVED3[24U]
```

#### 4.15.1.10 RESERVED4

```
uint32_t NVIC_Type::RESERVED4[56U]
```

#### 4.15.1.11 RESERVED5

```
uint32_t NVIC_Type::RESERVED5[644U]
```

#### 4.15.1.12 RSRVED1

```
uint32_t NVIC_Type::RSERVED1[24U]
```

#### 4.15.1.13 STIR

```
volatile uint32_t NVIC_Type::STIR
```

Offset: 0xE00 ( /W) Software Trigger Interrupt Register

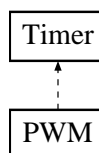
The documentation for this struct was generated from the following file:

- C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/[Micro.h](#)

## 4.16 PWM Class Reference

```
#include <PWM.h>
```

Inheritance diagram for PWM:



### Public Member Functions

- [PWM](#) ()
- [PWM](#) ([TIM\\_TypeDef](#) \*[TIM](#), [TIM\\_CHANNELS](#) channel, [TIM\\_REMAP](#) PWMremap)
- [PWM](#) ([PWM](#) \*pwm)
- void [PWMWrite](#) (float value)
- float [getWriedPWM](#) ()

### Additional Inherited Members

#### 4.16.1 Constructor & Destructor Documentation

#### 4.16.1.1 PWM() [1/3]

```
PWM::PWM ( ) [inline]
```

#### 4.16.1.2 PWM() [2/3]

```
PWM::PWM (
    TIM_TypeDef * TIM,
    TIM_CHANNELS channel,
    TIM_REMAP PWMremap )
```

#### 4.16.1.3 PWM() [3/3]

```
PWM::PWM (
    PWM * pwm )
```

### 4.16.2 Member Function Documentation

#### 4.16.2.1 getWritedPWM()

```
float PWM::getWritedPWM ( )
```

#### 4.16.2.2 PWMWrite()

```
void PWM::PWMWrite (
    float value )
```

The documentation for this class was generated from the following files:

- C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/[PWM.h](#)
- C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/[PWM.cpp](#)

## 4.17 PWR\_TypeDef Struct Reference

```
#include <Micro.h>
```

## Public Attributes

- volatile uint32\_t [CR](#)
- volatile uint32\_t [CSR](#)

### 4.17.1 Member Data Documentation

#### 4.17.1.1 CR

```
volatile uint32_t PWR_TypeDef::CR
```

#### 4.17.1.2 CSR

```
volatile uint32_t PWR_TypeDef::CSR
```

The documentation for this struct was generated from the following file:

- C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/[Micro.h](#)

## 4.18 RCC\_TypeDef Struct Reference

```
#include <Micro.h>
```

## Public Attributes

- volatile uint32\_t [CR](#)
- volatile uint32\_t [CFGR](#)
- volatile uint32\_t [CIR](#)
- volatile uint32\_t [APB2RSTR](#)
- volatile uint32\_t [APB1RSTR](#)
- volatile uint32\_t [AHBENR](#)
- volatile uint32\_t [APB2ENR](#)
- volatile uint32\_t [APB1ENR](#)
- volatile uint32\_t [BDCR](#)
- volatile uint32\_t [CSR](#)

### 4.18.1 Member Data Documentation

#### 4.18.1.1 AHBENR

```
volatile uint32_t RCC_TypeDef::AHBENR
```

#### 4.18.1.2 APB1ENR

```
volatile uint32_t RCC_TypeDef::APB1ENR
```

#### 4.18.1.3 APB1RSTR

```
volatile uint32_t RCC_TypeDef::APB1RSTR
```

#### 4.18.1.4 APB2ENR

```
volatile uint32_t RCC_TypeDef::APB2ENR
```

#### 4.18.1.5 APB2RSTR

```
volatile uint32_t RCC_TypeDef::APB2RSTR
```

#### 4.18.1.6 BDCR

```
volatile uint32_t RCC_TypeDef::BDCR
```

#### 4.18.1.7 CFGR

```
volatile uint32_t RCC_TypeDef::CFGR
```

#### 4.18.1.8 CIR

```
volatile uint32_t RCC_TypeDef::CIR
```

#### 4.18.1.9 CR

```
volatile uint32_t RCC_TypeDef::CR
```

#### 4.18.1.10 CSR

```
volatile uint32_t RCC_TypeDef::CSR
```

The documentation for this struct was generated from the following file:

- C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/[Micro.h](#)

## 4.19 Reflectance\_Sensor Class Reference

```
#include <REFLECTANCE_SENSOR.h>
```

### Public Member Functions

- [Reflectance\\_Sensor](#) ([Reflectance\\_Sensor](#) \*Sensor)
- [Reflectance\\_Sensor](#) ([ADC\\_CHANNELS](#) ADCChannel)
- void [Calib\\_Reflectance\\_Sensor](#) ()
- float [Reflectance\\_Read](#) ()

### Protected Member Functions

- [Reflectance\\_Sensor](#) ()

### Friends

- class [Line\\_Sensor](#)

## 4.19.1 Constructor & Destructor Documentation

### 4.19.1.1 Reflectance\_Sensor() [1/3]

```
Reflectance_Sensor::Reflectance_Sensor ( ) [protected]
```

#### 4.19.1.2 Reflectance\_Sensor() [2/3]

```
Reflectance_Sensor::Reflectance_Sensor (
    Reflectance_Sensor * Sensor )
```

#### 4.19.1.3 Reflectance\_Sensor() [3/3]

```
Reflectance_Sensor::Reflectance_Sensor (
    ADC_CHANNELS ADCChannel )
```

### 4.19.2 Member Function Documentation

#### 4.19.2.1 Calib\_Reflectance\_Sensor()

```
void Reflectance_Sensor::Calib_Reflectance_Sensor ( )
```

#### 4.19.2.2 Reflectance\_Read()

```
float Reflectance_Sensor::Reflectance_Read ( )
```

### 4.19.3 Friends And Related Function Documentation

#### 4.19.3.1 Line\_Sensor

```
friend class Line_Sensor [friend]
```

The documentation for this class was generated from the following files:

- C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/[REFLECTANCE\\_SENSOR.h](#)
- C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/[REFLECTANCE\\_SENSOR.cpp](#)

## 4.20 RTC\_TypeDef Struct Reference

```
#include <Micro.h>
```



## Public Attributes

- volatile uint16\_t [CRH](#)
- uint16\_t [RESERVED0](#)
- volatile uint16\_t [CRL](#)
- uint16\_t [RESERVED1](#)
- volatile uint16\_t [PRLH](#)
- uint16\_t [RESERVED2](#)
- volatile uint16\_t [PRL](#)
- uint16\_t [RESERVED3](#)
- volatile uint16\_t [DIVH](#)
- uint16\_t [RESERVED4](#)
- volatile uint16\_t [DIVL](#)
- uint16\_t [RESERVED5](#)
- volatile uint16\_t [CNTH](#)
- uint16\_t [RESERVED6](#)
- volatile uint16\_t [CNTL](#)
- uint16\_t [RESERVED7](#)
- volatile uint16\_t [ALRH](#)
- uint16\_t [RESERVED8](#)
- volatile uint16\_t [ALRL](#)
- uint16\_t [RESERVED9](#)

### 4.20.1 Member Data Documentation

#### 4.20.1.1 ALRH

```
volatile uint16_t RTC_TypeDef::ALRH
```

#### 4.20.1.2 ALRL

```
volatile uint16_t RTC_TypeDef::ALRL
```

#### 4.20.1.3 CNTH

```
volatile uint16_t RTC_TypeDef::CNTH
```

#### 4.20.1.4 CNTL

```
volatile uint16_t RTC_TypeDef::CNTL
```

#### 4.20.1.5 CRH

```
volatile uint16_t RTC_TypeDef::CRH
```

#### 4.20.1.6 CRL

```
volatile uint16_t RTC_TypeDef::CRL
```

#### 4.20.1.7 DIVH

```
volatile uint16_t RTC_TypeDef::DIVH
```

#### 4.20.1.8 DIVL

```
volatile uint16_t RTC_TypeDef::DIVL
```

#### 4.20.1.9 PRLH

```
volatile uint16_t RTC_TypeDef::PRLH
```

#### 4.20.1.10 PRL

```
volatile uint16_t RTC_TypeDef::PRL
```

#### 4.20.1.11 RESERVED0

```
uint16_t RTC_TypeDef::RESERVED0
```

#### 4.20.1.12 RESERVED1

```
uint16_t RTC_TypeDef::RESERVED1
```

#### 4.20.1.13 RESERVED2

```
uint16_t RTC_TypeDef::RESERVED2
```

#### 4.20.1.14 RESERVED3

```
uint16_t RTC_TypeDef::RESERVED3
```

#### 4.20.1.15 RESERVED4

```
uint16_t RTC_TypeDef::RESERVED4
```

#### 4.20.1.16 RESERVED5

```
uint16_t RTC_TypeDef::RESERVED5
```

#### 4.20.1.17 RESERVED6

```
uint16_t RTC_TypeDef::RESERVED6
```

#### 4.20.1.18 RESERVED7

```
uint16_t RTC_TypeDef::RESERVED7
```

#### 4.20.1.19 RESERVED8

```
uint16_t RTC_TypeDef::RESERVED8
```

#### 4.20.1.20 RESERVED9

```
uint16_t RTC_TypeDef::RESERVED9
```

The documentation for this struct was generated from the following file:

- C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/[Micro.h](#)

## 4.21 SysClock Class Reference

```
#include <SysClock.h>
```

### Public Member Functions

- [SysClock](#) ()
- void [SysClockInit](#) ()
- void [MCO](#) ()
- void [SysTickInit](#) ([SysTickBaseTimeEnum](#) BASE\_TIME)
- bool [SysTickGetEvent](#) ()

### 4.21.1 Constructor & Destructor Documentation

#### 4.21.1.1 SysClock()

```
SysClock::SysClock ( )
```

### 4.21.2 Member Function Documentation

#### 4.21.2.1 MCO()

```
void SysClock::MCO ( )
```

< Peripheral base address in the alias region

#### 4.21.2.2 SysClockInit()

```
void SysClock::SysClockInit ( )
```

< Peripheral base address in the alias region  
 < Peripheral base address in the alias region  
 < Peripheral base address in the alias region  
 < Peripheral base address in the alias region  
 < Peripheral base address in the alias region  
 < Peripheral base address in the alias region  
 < Peripheral base address in the alias region  
 < Peripheral base address in the alias region  
 < Peripheral base address in the alias region  
 < Flash registers base address  
 < Peripheral base address in the alias region  
 < Peripheral base address in the alias region

#### 4.21.2.3 SysTickGetEvent()

```
bool SysClock::SysTickGetEvent ( )
```

< System Control Space Base Address  
 < SysTick Base Address  
 < SysTick configuration struct

#### 4.21.2.4 SysTickInit()

```
void SysClock::SysTickInit (
    SysTickBaseTimeEnum BASE_TIME )
```

< System Control Space Base Address  
 < SysTick Base Address  
 < SysTick configuration struct  
 < System Control Space Base Address  
 < SysTick Base Address  
 < SysTick configuration struct

The documentation for this class was generated from the following files:

- C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/SysClock.h
- C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/SysClock.cpp

## 4.22 SysTick\_Type Struct Reference

```
#include <Micro.h>
```

### Public Attributes

- volatile uint32\_t [CTRL](#)
- volatile uint32\_t [LOAD](#)
- volatile uint32\_t [VAL](#)
- volatile uint32\_t [CALIB](#)

### 4.22.1 Member Data Documentation

#### 4.22.1.1 CALIB

```
volatile uint32_t SysTick_Type::CALIB
```

Offset: 0x00C (R/ ) SysTick Calibration Register

#### 4.22.1.2 CTRL

```
volatile uint32_t SysTick_Type::CTRL
```

Offset: 0x000 (R/W) SysTick Control and Status Register

#### 4.22.1.3 LOAD

```
volatile uint32_t SysTick_Type::LOAD
```

Offset: 0x004 (R/W) SysTick Reload Value Register

#### 4.22.1.4 VAL

```
volatile uint32_t SysTick_Type::VAL
```

Offset: 0x008 (R/W) SysTick Current Value Register

The documentation for this struct was generated from the following file:

- C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/[Micro.h](#)

## 4.23 TIM\_TypeDef Struct Reference

```
#include <Micro.h>
```

### Public Attributes

- volatile uint16\_t CR1
- uint16\_t RESERVED0
- volatile uint16\_t CR2
- uint16\_t RESERVED1
- volatile uint16\_t SMCR
- uint16\_t RESERVED2
- volatile uint16\_t DIER
- uint16\_t RESERVED3
- volatile uint16\_t SR
- uint16\_t RESERVED4
- volatile uint16\_t EGR
- uint16\_t RESERVED5
- volatile uint16\_t CCMR1
- uint16\_t RESERVED6
- volatile uint16\_t CCMR2
- uint16\_t RESERVED7
- volatile uint16\_t CCER
- uint16\_t RESERVED8
- volatile uint16\_t CNT
- uint16\_t RESERVED9
- volatile uint16\_t PSC
- uint16\_t RESERVED10
- volatile uint16\_t ARR
- uint16\_t RESERVED11
- volatile uint16\_t RCR
- uint16\_t RESERVED12
- volatile uint16\_t CCR1
- uint16\_t RESERVED13
- volatile uint16\_t CCR2
- uint16\_t RESERVED14
- volatile uint16\_t CCR3
- uint16\_t RESERVED15
- volatile uint16\_t CCR4
- uint16\_t RESERVED16
- volatile uint16\_t BDTR
- uint16\_t RESERVED17
- volatile uint16\_t DCR
- uint16\_t RESERVED18
- volatile uint16\_t DMAR
- uint16\_t RESERVED19

### 4.23.1 Member Data Documentation

#### 4.23.1.1 ARR

```
volatile uint16_t TIM_TypeDef::ARR
```

#### 4.23.1.2 BDTR

```
volatile uint16_t TIM_TypeDef::BDTR
```

#### 4.23.1.3 CCER

```
volatile uint16_t TIM_TypeDef::CCER
```

#### 4.23.1.4 CCMR1

```
volatile uint16_t TIM_TypeDef::CCMR1
```

#### 4.23.1.5 CCMR2

```
volatile uint16_t TIM_TypeDef::CCMR2
```

#### 4.23.1.6 CCR1

```
volatile uint16_t TIM_TypeDef::CCR1
```

#### 4.23.1.7 CCR2

```
volatile uint16_t TIM_TypeDef::CCR2
```

#### 4.23.1.8 CCR3

```
volatile uint16_t TIM_TypeDef::CCR3
```



#### 4.23.1.9 CCR4

```
volatile uint16_t TIM_TypeDef::CCR4
```

#### 4.23.1.10 CNT

```
volatile uint16_t TIM_TypeDef::CNT
```

#### 4.23.1.11 CR1

```
volatile uint16_t TIM_TypeDef::CR1
```

#### 4.23.1.12 CR2

```
volatile uint16_t TIM_TypeDef::CR2
```

#### 4.23.1.13 DCR

```
volatile uint16_t TIM_TypeDef::DCR
```

#### 4.23.1.14 DIER

```
volatile uint16_t TIM_TypeDef::DIER
```

#### 4.23.1.15 DMAR

```
volatile uint16_t TIM_TypeDef::DMAR
```

#### 4.23.1.16 EGR

```
volatile uint16_t TIM_TypeDef::EGR
```

**4.23.1.17 PSC**

```
volatile uint16_t TIM_TypeDef::PSC
```

**4.23.1.18 RCR**

```
volatile uint16_t TIM_TypeDef::RCR
```

**4.23.1.19 RESERVED0**

```
uint16_t TIM_TypeDef::RESERVED0
```

**4.23.1.20 RESERVED1**

```
uint16_t TIM_TypeDef::RESERVED1
```

**4.23.1.21 RESERVED10**

```
uint16_t TIM_TypeDef::RESERVED10
```

**4.23.1.22 RESERVED11**

```
uint16_t TIM_TypeDef::RESERVED11
```

**4.23.1.23 RESERVED12**

```
uint16_t TIM_TypeDef::RESERVED12
```

**4.23.1.24 RESERVED13**

```
uint16_t TIM_TypeDef::RESERVED13
```

**4.23.1.25 RESERVED14**

```
uint16_t TIM_TypeDef::RESERVED14
```

**4.23.1.26 RESERVED15**

```
uint16_t TIM_TypeDef::RESERVED15
```

**4.23.1.27 RESERVED16**

```
uint16_t TIM_TypeDef::RESERVED16
```

**4.23.1.28 RESERVED17**

```
uint16_t TIM_TypeDef::RESERVED17
```

**4.23.1.29 RESERVED18**

```
uint16_t TIM_TypeDef::RESERVED18
```

**4.23.1.30 RESERVED19**

```
uint16_t TIM_TypeDef::RESERVED19
```

**4.23.1.31 RESERVED2**

```
uint16_t TIM_TypeDef::RESERVED2
```

**4.23.1.32 RESERVED3**

```
uint16_t TIM_TypeDef::RESERVED3
```

**4.23.1.33 RESERVED4**

```
uint16_t TIM_TypeDef::RESERVED4
```

**4.23.1.34 RESERVED5**

```
uint16_t TIM_TypeDef::RESERVED5
```

**4.23.1.35 RESERVED6**

```
uint16_t TIM_TypeDef::RESERVED6
```

**4.23.1.36 RESERVED7**

```
uint16_t TIM_TypeDef::RESERVED7
```

**4.23.1.37 RESERVED8**

```
uint16_t TIM_TypeDef::RESERVED8
```

**4.23.1.38 RESERVED9**

```
uint16_t TIM_TypeDef::RESERVED9
```

**4.23.1.39 SMCR**

```
volatile uint16_t TIM_TypeDef::SMCR
```

## 4.23.1.40 SR

```
volatile uint16_t TIM_TypeDef::SR
```

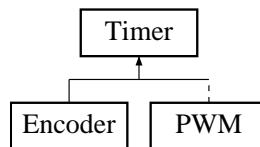
The documentation for this struct was generated from the following file:

- C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/[Micro.h](#)

## 4.24 Timer Class Reference

```
#include <TIMER.h>
```

Inheritance diagram for Timer:



### Public Member Functions

- [Timer](#) ([TIM\\_TypeDef](#) \*TIM, [TIM\\_CHANNELS](#) TIMChannel, [TIM\\_MODE](#) TIMMode)
- [Timer](#) ([TIM\\_TypeDef](#) \*TIM, [TIM\\_MODE](#) TIMMode)
- void [SetTIMRemap](#) ([TIM\\_REMAP](#) TIMRemap)
- [TIM\\_TypeDef](#) \* [GetTim](#) ()
- [TIM\\_CHANNELS](#) [GetTIMChannel](#) ()
- [TIM\\_MODE](#) [GetTIMMode](#) ()
- [TIM\\_REMAP](#) [GetTIMRemap](#) ()
- void [TimerInit](#) ()
- void [InterruptTime](#) (uint16\_t time)
- void [ClearInterruptFlag](#) ()

### Static Public Member Functions

- static uint32\_t [GetTime\\_usec](#) ()
- static uint32\_t [GetTime\\_milise](#) ()
- static void [delay](#) (uint32\_t delayTime\_usec)
- static void [Timer\\_Initialize](#) ()
- static void [Timer\\_Handler](#) ()
- static bool [verifyTimeInterrupt](#) ()

### Protected Member Functions

- [Timer](#) ()
- void [SetTim](#) ([TIM\\_TypeDef](#) \*TIM)
- void [SetTIMChannel](#) ([TIM\\_CHANNELS](#) TIMChannel)
- void [SetTIMMode](#) ([TIM\\_MODE](#) TIMMode)

## Protected Attributes

- [TIM\\_TypeDef](#) \* *TIM*

## 4.24.1 Constructor & Destructor Documentation

### 4.24.1.1 Timer() [1/3]

```
Timer::Timer ( ) [inline], [protected]
```

### 4.24.1.2 Timer() [2/3]

```
Timer::Timer (
    TIM\_TypeDef * TIM,
    TIM\_CHANNELS TIMChannel,
    TIM\_MODE TIMMode )
```

### 4.24.1.3 Timer() [3/3]

```
Timer::Timer (
    TIM\_TypeDef * TIM,
    TIM\_MODE TIMMode )
```

## 4.24.2 Member Function Documentation

### 4.24.2.1 ClearInterruptFlag()

```
void Timer::ClearInterruptFlag ( )
```

### 4.24.2.2 delay()

```
void Timer::delay (
    uint32_t delayTime_usec ) [static]
```

#### 4.24.2.3 GetTim()

```
TIM_TypeDef * Timer::GetTim ( )
```

#### 4.24.2.4 GetTIMChannel()

```
TIM_CHANNELS Timer::GetTIMChannel ( )
```

#### 4.24.2.5 GetTime\_miliseC()

```
uint32_t Timer::GetTime_miliseC ( ) [static]
```

#### 4.24.2.6 GetTime\_usec()

```
uint32_t Timer::GetTime_usec ( ) [static]
```

#### 4.24.2.7 GetTIMMode()

```
TIM_MODE Timer::GetTIMMode ( )
```

#### 4.24.2.8 GetTIMRemap()

```
TIM_REMAP Timer::GetTIMRemap ( )
```

< Peripheral base address in the alias region

< Peripheral base address in the alias region

< Peripheral base address in the alias region

< Peripheral base address in the alias region

#### 4.24.2.9 InterruptTime()

```
void Timer::InterruptTime (
    uint16_t time )
```

#### 4.24.2.10 SetTim()

```
void Timer::SetTim (
    TIM_TypeDef * TIM ) [protected]
```

#### 4.24.2.11 SetTIMChannel()

```
void Timer::SetTIMChannel (
    TIM_CHANNELS TIMChannel ) [protected]
```

#### 4.24.2.12 SetTIMMode()

```
void Timer::SetTIMMode (
    TIM_MODE TIMMode ) [protected]
```

#### 4.24.2.13 SetTIMRemap()

```
void Timer::SetTIMRemap (
    TIM_REMAP TIMRemap )
```

< Peripheral base address in the alias region

< Peripheral base address in the alias region

< Peripheral base address in the alias region

< Peripheral base address in the alias region

#### 4.24.2.14 Timer\_Handler()

```
void Timer::Timer_Handler ( ) [static]
```

#### 4.24.2.15 Timer\_Initiallize()

```
void Timer::Timer_Initiallize ( ) [static]
```



#### 4.24.2.16 TimerInit()

```
void Timer::TimerInit ( )
```

< Peripheral base address in the alias region

< Peripheral base address in the alias region

< Peripheral base address in the alias region

< Peripheral base address in the alias region

< Peripheral base address in the alias region

< Peripheral base address in the alias region

< Peripheral base address in the alias region

< Peripheral base address in the alias region

< Peripheral base address in the alias region

< System Control Space Base Address

< NVIC Base Address

< NVIC configuration struct

< Peripheral base address in the alias region

< System Control Space Base Address

< NVIC Base Address

< NVIC configuration struct

< Peripheral base address in the alias region

< System Control Space Base Address

< NVIC Base Address

< NVIC configuration struct

< Peripheral base address in the alias region

< System Control Space Base Address

< NVIC Base Address

< NVIC configuration struct

#### 4.24.2.17 verifyTimeInterrupt()

```
bool Timer::verifyTimeInterrupt ( ) [static]
```

### 4.24.3 Member Data Documentation

#### 4.24.3.1 TIM

```
TIM_TypeDef* Timer::TIM [protected]
```

The documentation for this class was generated from the following files:

- C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/[TIMER.h](#)
- C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/[TIMER.cpp](#)

## 4.25 USART Class Reference

```
#include <USART.h>
```

### Public Member Functions

- [USART](#) ([USART\\_TypeDef](#) \*Usart, [BD\\_ENUM](#) Baud\_Rate)
- void [Send](#) (char value)
- void [Send\\_Vec\\_16](#) (uint16\_t \*ptVec, uint16\_t size)
- void [sendFloat](#) (float \*ptFloat)
- void [sendUInt32](#) (uint32\_t \*ptUInt32)
- void [sendUInt16](#) (uint16\_t \*ptUInt16)
- uint8\_t [Receive](#) ()
- bool [Available](#) ()

### 4.25.1 Constructor & Destructor Documentation

#### 4.25.1.1 USART()

```
USART::USART (
    USART_TypeDef * Usart,
    BD_ENUM Baud_Rate )
```

### 4.25.2 Member Function Documentation

#### 4.25.2.1 Available()

```
bool USART::Available ( )
```

#### 4.25.2.2 Receive()

```
uint8_t USART::Receive ( )
```

#### 4.25.2.3 Send()

```
void USART::Send (
    char value )
```

#### 4.25.2.4 Send\_Vec\_16()

```
void USART::Send_Vec_16 (
    uint16_t * ptVec,
    uint16_t size )
```

#### 4.25.2.5 sendFloat()

```
void USART::sendFloat (
    float * ptFloat )
```

#### 4.25.2.6 sendUint16()

```
void USART::sendUint16 (
    uint16_t * ptUint16 )
```

#### 4.25.2.7 sendUint32()

```
void USART::sendUint32 (
    uint32_t * ptUint32 )
```

The documentation for this class was generated from the following files:

- C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/[USART.h](#)
- C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/[USART.cpp](#)

## 4.26 USART\_TypeDef Struct Reference

Universal Synchronous Asynchronous Receiver Transmitter.

```
#include <Micro.h>
```

### Public Attributes

- volatile uint16\_t [SR](#)
- uint16\_t [RESERVED0](#)
- volatile uint16\_t [DR](#)
- uint16\_t [RESERVED1](#)
- volatile uint16\_t [BRR](#)
- uint16\_t [RESERVED2](#)
- volatile uint16\_t [CR1](#)
- uint16\_t [RESERVED3](#)
- volatile uint16\_t [CR2](#)
- uint16\_t [RESERVED4](#)
- volatile uint16\_t [CR3](#)
- uint16\_t [RESERVED5](#)
- volatile uint16\_t [GTPR](#)
- uint16\_t [RESERVED6](#)

### 4.26.1 Detailed Description

Universal Synchronous Asynchronous Receiver Transmitter.

### 4.26.2 Member Data Documentation

#### 4.26.2.1 BRR

```
volatile uint16_t USART_TypeDef::BRR
```

#### 4.26.2.2 CR1

```
volatile uint16_t USART_TypeDef::CR1
```

#### 4.26.2.3 CR2

```
volatile uint16_t USART_TypeDef::CR2
```

#### 4.26.2.4 CR3

```
volatile uint16_t USART_TypeDef::CR3
```

#### 4.26.2.5 DR

```
volatile uint16_t USART_TypeDef::DR
```

#### 4.26.2.6 GTPR

```
volatile uint16_t USART_TypeDef::GTPR
```

#### 4.26.2.7 RESERVED0

```
uint16_t USART_TypeDef::RESERVED0
```

#### 4.26.2.8 RESERVED1

```
uint16_t USART_TypeDef::RESERVED1
```

#### 4.26.2.9 RESERVED2

```
uint16_t USART_TypeDef::RESERVED2
```

#### 4.26.2.10 RESERVED3

```
uint16_t USART_TypeDef::RESERVED3
```

#### 4.26.2.11 RESERVED4

```
uint16_t USART_TypeDef::RESERVED4
```

#### 4.26.2.12 RESERVED5

```
uint16_t USART_TypeDef::RESERVED5
```

#### 4.26.2.13 RESERVED6

```
uint16_t USART_TypeDef::RESERVED6
```

#### 4.26.2.14 SR

```
volatile uint16_t USART_TypeDef::SR
```

The documentation for this struct was generated from the following file:

- C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/[Micro.h](#)

## Chapter 5

# File Documentation

### 5.1 C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/ADC.cpp File Reference

```
#include "ADC.h"
```

### 5.2 C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/ADC.h File Reference

```
#include "GPIO.h"  
#include "TIMER.h"
```

#### Classes

- class [ADC](#)

#### Enumerations

- enum [ADC\\_CONVERSION\\_MODES](#) { [SINGLE\\_CONVERSION](#), [CONTINUOUS\\_CONVERSION](#) }
- enum [ADC\\_CHANNELS](#) {  
    [ADC\\_CH0](#) = 0, [ADC\\_CH1](#), [ADC\\_CH2](#), [ADC\\_CH3](#),  
    [ADC\\_CH4](#), [ADC\\_CH5](#), [ADC\\_CH6](#), [ADC\\_CH7](#),  
    [ADC\\_CH8](#), [ADC\\_CH9](#) }

#### 5.2.1 Enumeration Type Documentation

##### 5.2.1.1 ADC\_CHANNELS

```
enum ADC\_CHANNELS
```

**Enumerator**

ADC_CH0	
ADC_CH1	
ADC_CH2	
ADC_CH3	
ADC_CH4	
ADC_CH5	
ADC_CH6	
ADC_CH7	
ADC_CH8	
ADC_CH9	

**5.2.1.2 ADC\_CONVERSION\_MODES**

enum [ADC\\_CONVERSION\\_MODES](#)

**Enumerator**

SINGLE_CONVERSION	
CONTINUOUS_CONVERSION	

### 5.3 C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/ANGLE\_LINE\_SENSOR.cpp File Reference

### 5.4 C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/ANGLE\_LINE\_SENSOR.h File Reference

```
#include "LINE_SENSOR.h"
```

**Classes**

- class [ANGLE\\_LINE\\_SENSOR\\_H](#)

### 5.5 C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/ENCODER.cpp File Reference

```
#include "ENCODER.h"
```



## 5.6 C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/ENCODER.h File Reference

```
#include "GPIO.h"
#include "TIMER.h"
```

### Classes

- class [Encoder](#)

### Macros

- #define [Max\\_delay\\_Ticks\\_Time](#) 50000
- #define [AutoReload\\_Ticks](#) 1
- #define [Ticks\\_till\\_int](#) ([AutoReload\\_Ticks](#)+1)
- #define [encoderFilterOrder](#) 6
- #define [pi](#) 3.14159265359
- #define [pulsePerRevolution](#) 120
- #define [ticksToRad](#) 2\*pi/[pulsePerRevolution](#)

### Enumerations

- enum [ENCODER\\_ENUM](#) {  
    [Encoder\\_TIM1](#) = 0, [Encoder\\_TIM2](#), [Encoder\\_TIM3](#), [Encoder\\_TIM4](#),  
    [NUMBER\\_OF\\_ENCODERS](#) }
- enum [ENCODER\\_DIRECTION](#) { [forward](#), [backward](#) }

## 5.6.1 Macro Definition Documentation

### 5.6.1.1 AutoReload\_Ticks

```
#define AutoReload_Ticks 1
```

### 5.6.1.2 encoderFilterOrder

```
#define encoderFilterOrder 6
```

### 5.6.1.3 Max\_delay\_Ticks\_Time

```
#define Max_delay_Ticks_Time 50000
```

### 5.6.1.4 pi

```
#define pi 3.14159265359
```

### 5.6.1.5 pulsePerRevolution

```
#define pulsePerRevolution 120
```

### 5.6.1.6 Ticks\_till\_int

```
#define Ticks_till_int (AutoReload_Ticks+1)
```

### 5.6.1.7 ticksToRad

```
#define ticksToRad 2*pi/pulsePerRevolution
```

## 5.6.2 Enumeration Type Documentation

### 5.6.2.1 ENCODER\_DIRECTION

```
enum ENCODER_DIRECTION
```

Enumerator

forward	
backward	

5.6.2.2 ENCODER\_ENUM

---

```
enum ENCODER_ENUM
```

## Enumerator

Encoder_TIM1	
Encoder_TIM2	
Encoder_TIM3	
Encoder_TIM4	
NUMBER_OF_ENCODERS	

## 5.7 C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/FOLLOWING\_CONTROLLER.cpp File Reference

```
#include "FOLLOWING_CONTROLLER.h"
```

## 5.8 C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/FOLLOWING\_CONTROLLER.h File Reference

```
#include "KINEMATIC_CONTROL.h"
#include "math.h"
```

### Classes

- class [Controller](#)

### Macros

- #define [Kx](#) 60
- #define [Ky](#) 900
- #define [Kteta](#) 20
- #define [K4](#) 400
- #define [K1](#) [Kx](#)
- #define [K2](#) [Ky](#)
- #define [K3](#) [Kz](#)
- #define [Kxk](#) 1
- #define [Kyk](#) 200
- #define [Ktetak](#) 0.01
- #define [K4k](#) 0.001
- #define [K1a](#) 22
- #define [K2a](#) 1000
- #define [v\\_ref\\_ini](#) 0.5
- #define [w\\_ref\\_ini](#) 0

## 5.8.1 Macro Definition Documentation

### 5.8.1.1 K1

```
#define K1 Kx
```

### 5.8.1.2 K1a

```
#define K1a 22
```

### 5.8.1.3 K2

```
#define K2 Ky
```

### 5.8.1.4 K2a

```
#define K2a 1000
```

### 5.8.1.5 K3

```
#define K3 Kz
```

### 5.8.1.6 K4

```
#define K4 400
```

### 5.8.1.7 K4k

```
#define K4k 0.001
```

#### 5.8.1.8 Kteta

```
#define Kteta 20
```

#### 5.8.1.9 Ktetak

```
#define Ktetak 0.01
```

#### 5.8.1.10 Kx

```
#define Kx 60
```

#### 5.8.1.11 Kxk

```
#define Kxk 1
```

#### 5.8.1.12 Ky

```
#define Ky 900
```

#### 5.8.1.13 Kyk

```
#define Kyk 200
```

#### 5.8.1.14 v\_ref\_ini

```
#define v_ref_ini 0.5
```

#### 5.8.1.15 w\_ref\_ini

```
#define w_ref_ini 0
```

## 5.9 C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/GPIO.cpp File Reference

```
#include "Micro.h"
#include "GPIO.h"
```

## 5.10 C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/GPIO.h File Reference

```
#include "Micro.h"
```

### Classes

- struct [GPIO\\_STRUCT](#)
- class [GPIO](#)

### Macros

- #define [HIGH](#) 1
- #define [LOW](#) 0

### Enumerations

- enum [GPIO\\_MODES](#) {  
[INPUT\\_ANALOG](#) = 0x0, [GP\\_OUTPUT\\_PUSH\\_PULL\\_10MHZ](#) = 0x1, [GP\\_OUTPUT\\_PUSH\\_PULL\\_2MHZ](#) = 0x2, [GP\\_OUTPUT\\_PUSH\\_PULL\\_50MHZ](#) = 0x3,  
[INPUT\\_FLOATING](#) = 0x4, [GP\\_OUTPUT\\_OPEN\\_DRAIN\\_10MHZ](#) = 0x5, [GP\\_OUTPUT\\_OPEN\\_DRAIN\\_2MHZ](#) = 0x6, [GP\\_OUTPUT\\_OPEN\\_DRAIN\\_50MHZ](#) = 0x7,  
[INPUT\\_PULL\\_UP\\_DOWN](#) = 0x8, [AF\\_OUTPUT\\_PUSH\\_PULL\\_10MHZ](#) = 0x9, [AF\\_OUTPUT\\_PUSH\\_PULL\\_2MHZ](#) = 0xA, [AF\\_OUTPUT\\_PUSH\\_PULL\\_50MHZ](#) = 0xB,  
[AF\\_OUTPUT\\_OPEN\\_DRAIN\\_10MHZ](#) = 0xD, [AF\\_OUTPUT\\_OPEN\\_DRAIN\\_2MHZ](#) = 0xE, [AF\\_OUTPUT\\_OPEN\\_DRAIN\\_50MHZ](#) = 0xF }
- enum [PIN\\_NUMBERS](#) {  
[PIN0](#) = 0, [PIN1](#), [PIN2](#), [PIN3](#),  
[PIN4](#), [PIN5](#), [PIN6](#), [PIN7](#),  
[PIN8](#), [PIN9](#), [PIN10](#), [PIN11](#),  
[PIN12](#), [PIN13](#), [PIN14](#), [PIN15](#),  
[NUMBER\\_OF\\_PINS](#) }
- enum [PU\\_PD\\_ENUM](#) { [PULL\\_DOWN](#) = 0, [PULL\\_UP](#) = 1 }
- enum [GPIO\\_IO\\_ENUM](#) {  
[PA0](#), [PA1](#), [PA2](#), [PA3](#),  
[PA4](#), [PA5](#), [PA6](#), [PA7](#),  
[PA8](#), [PA9](#), [PA10](#), [PA11](#),  
[PA12](#), [PA15](#), [PB0](#), [PB1](#),  
[PB3](#), [PB4](#), [PB5](#), [PB6](#),  
[PB7](#), [PB8](#), [PB9](#), [PB10](#),  
[PB11](#), [PB12](#), [PB13](#), [PB14](#),  
[PB15](#), [PC13](#), [PC14](#), [PC15](#),  
[NUM\\_OF\\_IOS](#) }

## 5.10.1 Macro Definition Documentation

### 5.10.1.1 HIGH

```
#define HIGH 1
```

### 5.10.1.2 LOW

```
#define LOW 0
```

## 5.10.2 Enumeration Type Documentation

### 5.10.2.1 GPIO\_IO\_ENUM

```
enum GPIO\_IO\_ENUM
```

#### Enumerator

PA0	
PA1	
PA2	
PA3	
PA4	
PA5	
PA6	
PA7	
PA8	
PA9	
PA10	
PA11	
PA12	
PA15	
PB0	
PB1	
PB3	
PB4	
PB5	
PB6	
PB7	
PB8	
PB9	
PB10	

## Enumerator

PB11	
PB12	
PB13	
PB14	
PB15	
PC13	
PC14	
PC15	
NUM_OF_IOS	

### 5.10.2.2 GPIO\_MODES

enum [GPIO\\_MODES](#)

## Enumerator

INPUT_ANALOG	
GP_OUTPUT_PUSH_PULL_10MHZ	
GP_OUTPUT_PUSH_PULL_2MHZ	
GP_OUTPUT_PUSH_PULL_50MHZ	
INPUT_FLOATING	
GP_OUTPUT_OPEN_DRAIN_10MHZ	
GP_OUTPUT_OPEN_DRAIN_2MHZ	
GP_OUTPUT_OPEN_DRAIN_50MHZ	
INPUT_PULL_UP_DOWN	
AF_OUTPUT_PUSH_PULL_10MHZ	
AF_OUTPUT_PUSH_PULL_2MHZ	
AF_OUTPUT_PUSH_PULL_50MHZ	
AF_OUTPUT_OPEN_DRAIN_10MHZ	
AF_OUTPUT_OPEN_DRAIN_2MHZ	
AF_OUTPUT_OPEN_DRAIN_50MHZ	

### 5.10.2.3 PIN\_NUMBERS

enum [PIN\\_NUMBERS](#)

## Enumerator

PIN0	
PIN1	
PIN2	
PIN3	
PIN4	



#### Enumerator

PIN5	
PIN6	
PIN7	
PIN8	
PIN9	
PIN10	
PIN11	
PIN12	
PIN13	
PIN14	
PIN15	
NUMBER_OF_PINS	

#### 5.10.2.4 PU\_PD\_ENUM

```
enum PU_PD_ENUM
```

#### Enumerator

PULL_DOWN	
PULL_UP	

## 5.11 C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/KINEMATIC\_CONTROL.cpp File Reference

```
#include "KINEMATIC_CONTROL.h"
```

## 5.12 C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/KINEMATIC\_CONTROL.h File Reference

```
#include "MOTOR.h"
#include "LINE_SENSOR.h"
#include "math.h"
```

### Classes

- class [Kinematic](#)

## Macros

- `#define L 0.069`
- `#define angleCorrection 1`
- `#define distanceCorrection 1`
- `#define angleFilterOrder 2`
- `#define integrationTime Time_between_int/1000000`
- `#define Kw 455.5`
- `#define Pw 12.97`
- `#define ts_w 0.05`
- `#define KDw 4/(Kw*ts_w)`
- `#define KPw Pw*KDw`

### 5.12.1 Macro Definition Documentation

#### 5.12.1.1 angleCorrection

```
#define angleCorrection 1
```

#### 5.12.1.2 angleFilterOrder

```
#define angleFilterOrder 2
```

#### 5.12.1.3 distanceCorrection

```
#define distanceCorrection 1
```

#### 5.12.1.4 integrationTime

```
#define integrationTime Time_between_int/1000000
```

#### 5.12.1.5 KDw

```
#define KDw 4/(Kw*ts_w)
```

#### 5.12.1.6 KPw

```
#define KPw Pw*KDw
```

#### 5.12.1.7 Kw

```
#define Kw 455.5
```

#### 5.12.1.8 L

```
#define L 0.069
```

#### 5.12.1.9 Pw

```
#define Pw 12.97
```

#### 5.12.1.10 ts\_w

```
#define ts_w 0.05
```

### 5.13 C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/line\_follower.cpp File Reference

### 5.14 C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/LINE\_SENSOR.cpp File Reference

```
#include "LINE_SENSOR.h"
```

## 5.15 C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/LINE\_SENSOR.h File Reference

```
#include "REFLECTANCE_SENSOR.h"
```

### Classes

- class [Line\\_Sensor](#)

### Macros

- #define [calib\\_sensores](#) 220
- #define [maxDistance](#) 0.035
- #define [sensorFilterOrder](#) 2

### 5.15.1 Macro Definition Documentation

#### 5.15.1.1 [calib\\_sensores](#)

```
#define calib_sensores 220
```

#### 5.15.1.2 [maxDistance](#)

```
#define maxDistance 0.035
```

#### 5.15.1.3 [sensorFilterOrder](#)

```
#define sensorFilterOrder 2
```

## 5.16 C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/main.cpp File Reference

## 5.17 C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/malha\_aberta.cpp File Reference

## 5.18 C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/malha\_fechada.cpp File Reference

```
#include "Micro.h"
#include "SysClock.h"
#include "GPIO.h"
#include "TIMER.h"
#include "PWM.h"
#include "ENCODER.h"
#include "ADC.h"
#include "REFLECTANCE_SENSOR.h"
#include "LINE_SENSOR.h"
#include "MOTOR.h"
#include "USART.h"
#include "KINEMATIC_CONTROL.h"
#include "FOLLOWING_CONTROLER.h"
#include "SERIAL_COMMUNICATION.h"
#include "math.h"
```

### Macros

- #define `stepTime` 1000
- #define `steps` 7
- #define `testTime` `stepTime*steps`

### Functions

- int `main` ()
- void `TIM1_UP_IRQHandler` ()
- void `TIM2_IRQHandler` ()
- void `TIM3_IRQHandler` ()
- void `TIM4_IRQHandler` ()

## Variables

- float `xtest`
- float `ytest`
- float `vtest`
- float `wtest`
- float `pwmtest`
- float `linevalue`
- float `lineangle`
- uint32\_t `t_ini`
- uint32\_t `time`
- uint32\_t `delta_time`
- uint32\_t `time_test`
- uint8\_t `counter` = 0
- float `RPM_value` [12]

## 5.18.1 Macro Definition Documentation

### 5.18.1.1 steps

```
#define steps 7
```

### 5.18.1.2 stepTime

```
#define stepTime 1000
```

### 5.18.1.3 testTime

```
#define testTime stepTime*steps
```

## 5.18.2 Function Documentation

### 5.18.2.1 main()

```
int main ( )
```

< Peripheral base address in the alias region

< Peripheral base address in the alias region

< Peripheral base address in the alias region

< Peripheral base address in the alias region

< Peripheral base address in the alias region

< Peripheral base address in the alias region

### 5.18.2.2 TIM1\_UP\_IRQHandler()

```
void TIM1_UP_IRQHandler ( )
```

< Peripheral base address in the alias region

### 5.18.2.3 TIM2\_IRQHandler()

```
void TIM2_IRQHandler ( )
```

< Peripheral base address in the alias region

### 5.18.2.4 TIM3\_IRQHandler()

```
void TIM3_IRQHandler ( )
```

< Peripheral base address in the alias region

### 5.18.2.5 TIM4\_IRQHandler()

```
void TIM4_IRQHandler ( )
```

< Peripheral base address in the alias region

## 5.18.3 Variable Documentation

### 5.18.3.1 counter

```
uint8_t counter = 0
```

**5.18.3.2 delta\_time**

```
uint32_t delta_time
```

**5.18.3.3 lineangle**

```
float lineangle
```

**5.18.3.4 linevalue**

```
float linevalue
```

**5.18.3.5 pwmtest**

```
float pwmtest
```

**5.18.3.6 RPM\_value**

```
float RPM_value[12]
```

**5.18.3.7 t\_ini**

```
uint32_t t_ini
```

**5.18.3.8 time**

```
uint32_t time
```

**5.18.3.9 time\_test**

```
uint32_t time_test
```



#### 5.18.3.10 vtest

```
float vtest
```

#### 5.18.3.11 wtest

```
float wtest
```

#### 5.18.3.12 xtest

```
float xtest
```

#### 5.18.3.13 ytest

```
float ytest
```

## 5.19 C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/Micro.c File Reference

## 5.20 C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/Micro.h File Reference

```
#include "stdint.h"
```

### Classes

- struct [ADC\\_TypeDef](#)
- struct [RCC\\_TypeDef](#)
- struct [TIM\\_TypeDef](#)
- struct [USART\\_TypeDef](#)

*Universal Synchronous Asynchronous Receiver Transmitter.*

- struct [GPIO\\_TypeDef](#)
- struct [AFIO\\_TypeDef](#)
- struct [PWR\\_TypeDef](#)
- struct [RTC\\_TypeDef](#)
- struct [FLASH\\_TypeDef](#)
- struct [NVIC\\_Type](#)
- struct [SysTick\\_Type](#)

## Macros

- #define FLASH\_BASE ((uint32\_t)0x08000000)
- #define SRAM\_BASE ((uint32\_t)0x20000000)
- #define PERIPH\_BASE ((uint32\_t)0x40000000)
- #define SRAM\_BB\_BASE ((uint32\_t)0x22000000)
- #define PERIPH\_BB\_BASE ((uint32\_t)0x42000000)
- #define FSMC\_R\_BASE ((uint32\_t)0xA0000000)
- #define APB1PERIPH\_BASE PERIPH\_BASE
- #define APB2PERIPH\_BASE (PERIPH\_BASE + 0x10000)
- #define AHBPERIPH\_BASE (PERIPH\_BASE + 0x20000)
- #define TIM2\_BASE (APB1PERIPH\_BASE + 0x0000)
- #define TIM3\_BASE (APB1PERIPH\_BASE + 0x0400)
- #define TIM4\_BASE (APB1PERIPH\_BASE + 0x0800)
- #define RTC\_BASE (APB1PERIPH\_BASE + 0x2800)
- #define USART2\_BASE (APB1PERIPH\_BASE + 0x4400)
- #define USART3\_BASE (APB1PERIPH\_BASE + 0x4800)
- #define PWR\_BASE (APB1PERIPH\_BASE + 0x7000)
- #define AFIO\_BASE (APB2PERIPH\_BASE + 0x0000)
- #define GPIOA\_BASE (APB2PERIPH\_BASE + 0x0800)
- #define GPIOB\_BASE (APB2PERIPH\_BASE + 0x0C00)
- #define GPIOC\_BASE (APB2PERIPH\_BASE + 0x1000)
- #define GPIOD\_BASE (APB2PERIPH\_BASE + 0x1400)
- #define GPIOE\_BASE (APB2PERIPH\_BASE + 0x1800)
- #define GPIOF\_BASE (APB2PERIPH\_BASE + 0x1C00)
- #define GPIOG\_BASE (APB2PERIPH\_BASE + 0x2000)
- #define ADC1\_BASE (APB2PERIPH\_BASE + 0x2400)
- #define ADC2\_BASE (APB2PERIPH\_BASE + 0x2800)
- #define TIM1\_BASE (APB2PERIPH\_BASE + 0x2C00)
- #define USART1\_BASE (APB2PERIPH\_BASE + 0x3800)
- #define RCC\_BASE (AHBPERIPH\_BASE + 0x1000)
- #define FLASH\_R\_BASE (AHBPERIPH\_BASE + 0x2000)
- #define TIM2 ((TIM\_TypeDef \*) TIM2\_BASE)
- #define TIM3 ((TIM\_TypeDef \*) TIM3\_BASE)
- #define TIM4 ((TIM\_TypeDef \*) TIM4\_BASE)
- #define RTC ((RTC\_TypeDef \*) RTC\_BASE)
- #define USART2 ((USART\_TypeDef \*) USART2\_BASE)
- #define USART3 ((USART\_TypeDef \*) USART3\_BASE)
- #define AFIO ((AFIO\_TypeDef \*) AFIO\_BASE)
- #define GPIOA ((GPIO\_TypeDef \*) GPIOA\_BASE)
- #define GPIOB ((GPIO\_TypeDef \*) GPIOB\_BASE)
- #define GPIOC ((GPIO\_TypeDef \*) GPIOC\_BASE)
- #define GPIOD ((GPIO\_TypeDef \*) GPIOD\_BASE)
- #define GPIOE ((GPIO\_TypeDef \*) GPIOE\_BASE)
- #define GPIOF ((GPIO\_TypeDef \*) GPIOF\_BASE)
- #define GPIOG ((GPIO\_TypeDef \*) GPIOG\_BASE)
- #define ADC1 ((ADC\_TypeDef \*) ADC1\_BASE)
- #define ADC2 ((ADC\_TypeDef \*) ADC2\_BASE)
- #define TIM1 ((TIM\_TypeDef \*) TIM1\_BASE)
- #define RCC ((RCC\_TypeDef \*) RCC\_BASE)
- #define USART1 ((USART\_TypeDef \*) USART1\_BASE)
- #define PWR ((PWR\_TypeDef \*) PWR\_BASE)
- #define FLASH ((FLASH\_TypeDef \*) FLASH\_R\_BASE)
- #define SCS\_BASE (0xE000E000UL)
- #define ITM\_BASE (0xE0000000UL)

- #define [DWT\\_BASE](#) (0xE0001000UL)
- #define [TPI\\_BASE](#) (0xE0040000UL)
- #define [CoreDebug\\_BASE](#) (0xE000EDF0UL)
- #define [SysTick\\_BASE](#) ([SCS\\_BASE](#) + 0x0010UL)
- #define [NVIC\\_BASE](#) ([SCS\\_BASE](#) + 0x0100UL)
- #define [SCB\\_BASE](#) ([SCS\\_BASE](#) + 0x0D00UL)
- #define [SCnSCB](#) (([SCnSCB\\_Type](#) \*) [SCS\\_BASE](#) )
- #define [SCB](#) (([SCB\\_Type](#) \*) [SCB\\_BASE](#) )
- #define [SysTick](#) (([SysTick\\_Type](#) \*) [SysTick\\_BASE](#) )
- #define [NVIC](#) (([NVIC\\_Type](#) \*) [NVIC\\_BASE](#) )
- #define [ITM](#) (([ITM\\_Type](#) \*) [ITM\\_BASE](#) )
- #define [DWT](#) (([DWT\\_Type](#) \*) [DWT\\_BASE](#) )
- #define [TPI](#) (([TPI\\_Type](#) \*) [TPI\\_BASE](#) )
- #define [CoreDebug](#) (([CoreDebug\\_Type](#) \*) [CoreDebug\\_BASE](#))

## Typedefs

- typedef enum [IRQn](#) [IRQn\\_Type](#)

## Enumerations

- enum [IRQn](#) {  
[NonMaskableInt\\_IRQn](#) = -14, [MemoryManagement\\_IRQn](#) = -12, [BusFault\\_IRQn](#) = -11, [UsageFault\\_IRQn](#) = -10,  
[SVCall\\_IRQn](#) = -5, [DebugMonitor\\_IRQn](#) = -4, [PendSV\\_IRQn](#) = -2, [SysTick\\_IRQn](#) = -1,  
[WWDG\\_IRQn](#) = 0, [PVD\\_IRQn](#) = 1, [TAMPER\\_IRQn](#) = 2, [RTC\\_IRQn](#) = 3,  
[FLASH\\_IRQn](#) = 4, [RCC\\_IRQn](#) = 5, [EXTI0\\_IRQn](#) = 6, [EXTI1\\_IRQn](#) = 7,  
[EXTI2\\_IRQn](#) = 8, [EXTI3\\_IRQn](#) = 9, [EXTI4\\_IRQn](#) = 10, [DMA1\\_Channel1\\_IRQn](#) = 11,  
[DMA1\\_Channel2\\_IRQn](#) = 12, [DMA1\\_Channel3\\_IRQn](#) = 13, [DMA1\\_Channel4\\_IRQn](#) = 14, [DMA1\\_Channel5\\_IRQn](#) = 15,  
[DMA1\\_Channel6\\_IRQn](#) = 16, [DMA1\\_Channel7\\_IRQn](#) = 17, [ADC1\\_2\\_IRQn](#) = 18, [USB\\_HP\\_CAN1\\_TX\\_IRQn](#) = 19,  
[USB\\_LP\\_CAN1\\_RX0\\_IRQn](#) = 20, [CAN1\\_RX1\\_IRQn](#) = 21, [CAN1\\_SCE\\_IRQn](#) = 22, [EXTI9\\_5\\_IRQn](#) = 23,  
[TIM1\\_BRK\\_IRQn](#) = 24, [TIM1\\_UP\\_IRQn](#) = 25, [TIM1\\_TRG\\_COM\\_IRQn](#) = 26, [TIM1\\_CC\\_IRQn](#) = 27,  
[TIM2\\_IRQn](#) = 28, [TIM3\\_IRQn](#) = 29, [TIM4\\_IRQn](#) = 30, [I2C1\\_EV\\_IRQn](#) = 31,  
[I2C1\\_ER\\_IRQn](#) = 32, [I2C2\\_EV\\_IRQn](#) = 33, [I2C2\\_ER\\_IRQn](#) = 34, [SPI1\\_IRQn](#) = 35,  
[SPI2\\_IRQn](#) = 36, [USART1\\_IRQn](#) = 37, [USART2\\_IRQn](#) = 38, [USART3\\_IRQn](#) = 39,  
[EXTI15\\_10\\_IRQn](#) = 40, [RTCAlarm\\_IRQn](#) = 41, [USBWakeUp\\_IRQn](#) = 42 }

### 5.20.1 Macro Definition Documentation

#### 5.20.1.1 ADC1

```
#define ADC1 ((ADC\_TypeDef *) ADC1\_BASE)
```

### 5.20.1.2 ADC1\_BASE

```
#define ADC1_BASE (APB2PERIPH_BASE + 0x2400)
```

### 5.20.1.3 ADC2

```
#define ADC2 ((ADC_TypeDef *) ADC2_BASE)
```

### 5.20.1.4 ADC2\_BASE

```
#define ADC2_BASE (APB2PERIPH_BASE + 0x2800)
```

### 5.20.1.5 AFIO

```
#define AFIO ((AFIO_TypeDef *) AFIO_BASE)
```

### 5.20.1.6 AFIO\_BASE

```
#define AFIO_BASE (APB2PERIPH_BASE + 0x0000)
```

### 5.20.1.7 AHBPERIPH\_BASE

```
#define AHBPERIPH_BASE (PERIPH_BASE + 0x20000)
```

### 5.20.1.8 APB1PERIPH\_BASE

```
#define APB1PERIPH_BASE PERIPH_BASE
```

### 5.20.1.9 APB2PERIPH\_BASE

```
#define APB2PERIPH_BASE (PERIPH_BASE + 0x10000)
```

### 5.20.1.10 CoreDebug

```
#define CoreDebug ((CoreDebug_Type *) CoreDebug_BASE)
```

Core Debug configuration struct

### 5.20.1.11 CoreDebug\_BASE

```
#define CoreDebug_BASE (0xE000EDF0UL)
```

Core Debug Base Address

### 5.20.1.12 DWT

```
#define DWT ((DWT_Type *) DWT_BASE )
```

DWT configuration struct

### 5.20.1.13 DWT\_BASE

```
#define DWT_BASE (0xE0001000UL)
```

DWT Base Address

### 5.20.1.14 FLASH

```
#define FLASH ((FLASH_TypeDef *) FLASH_R_BASE)
```

### 5.20.1.15 FLASH\_BASE

```
#define FLASH_BASE ((uint32_t)0x08000000)
```

FLASH base address in the alias region

### 5.20.1.16 FLASH\_R\_BASE

```
#define FLASH_R_BASE (AHBPERIPH_BASE + 0x2000)
```

Flash registers base address

### 5.20.1.17 FSMC\_R\_BASE

```
#define FSMC_R_BASE ((uint32_t)0xA0000000)
```

FSMC registers base address Peripheral memory map

#### 5.20.1.18 GPIOA

```
#define GPIOA ((GPIO_TypeDef *) GPIOA_BASE)
```

#### 5.20.1.19 GPIOA\_BASE

```
#define GPIOA_BASE (APB2PERIPH_BASE + 0x0800)
```

#### 5.20.1.20 GPIOB

```
#define GPIOB ((GPIO_TypeDef *) GPIOB_BASE)
```

#### 5.20.1.21 GPIOB\_BASE

```
#define GPIOB_BASE (APB2PERIPH_BASE + 0x0C00)
```

#### 5.20.1.22 GPIOC

```
#define GPIOC ((GPIO_TypeDef *) GPIOC_BASE)
```

#### 5.20.1.23 GPIOC\_BASE

```
#define GPIOC_BASE (APB2PERIPH_BASE + 0x1000)
```

#### 5.20.1.24 GPIOD

```
#define GPIOD ((GPIO_TypeDef *) GPIOD_BASE)
```

#### 5.20.1.25 GPIOD\_BASE

```
#define GPIOD_BASE (APB2PERIPH_BASE + 0x1400)
```

### 5.20.1.26 GPIOE

```
#define GPIOE ((GPIO_TypeDef *) GPIOE_BASE)
```

### 5.20.1.27 GPIOE\_BASE

```
#define GPIOE_BASE (APB2PERIPH_BASE + 0x1800)
```

### 5.20.1.28 GPIOF

```
#define GPIOF ((GPIO_TypeDef *) GPIOF_BASE)
```

### 5.20.1.29 GPIOF\_BASE

```
#define GPIOF_BASE (APB2PERIPH_BASE + 0x1C00)
```

### 5.20.1.30 GPIOG

```
#define GPIOG ((GPIO_TypeDef *) GPIOG_BASE)
```

### 5.20.1.31 GPIOG\_BASE

```
#define GPIOG_BASE (APB2PERIPH_BASE + 0x2000)
```

### 5.20.1.32 ITM

```
#define ITM ((ITM_Type *) ITM_BASE )
```

ITM configuration struct

### 5.20.1.33 ITM\_BASE

```
#define ITM_BASE (0xE0000000UL)
```

ITM Base Address

#### 5.20.1.34 NVIC

```
#define NVIC ((NVIC_Type *) NVIC_BASE )
```

NVIC configuration struct

#### 5.20.1.35 NVIC\_BASE

```
#define NVIC_BASE (SCS_BASE + 0x0100UL)
```

NVIC Base Address

#### 5.20.1.36 PERIPH\_BASE

```
#define PERIPH_BASE ((uint32_t)0x40000000)
```

Peripheral base address in the alias region

#### 5.20.1.37 PERIPH\_BB\_BASE

```
#define PERIPH_BB_BASE ((uint32_t)0x42000000)
```

Peripheral base address in the bit-band region

#### 5.20.1.38 PWR

```
#define PWR ((PWR_TypeDef *) PWR_BASE)
```

#### 5.20.1.39 PWR\_BASE

```
#define PWR_BASE (APB1PERIPH_BASE + 0x7000)
```

#### 5.20.1.40 RCC

```
#define RCC ((RCC_TypeDef *) RCC_BASE)
```

#### 5.20.1.41 RCC\_BASE

```
#define RCC_BASE (AHBPERIPH_BASE + 0x1000)
```



#### 5.20.1.42 RTC

```
#define RTC ((RTC_TypeDef *) RTC_BASE)
```

#### 5.20.1.43 RTC\_BASE

```
#define RTC_BASE (APB1PERIPH_BASE + 0x2800)
```

#### 5.20.1.44 SCB

```
#define SCB ((SCB_Type *) SCB_BASE )
```

SCB configuration struct

#### 5.20.1.45 SCB\_BASE

```
#define SCB_BASE (SCS_BASE + 0x0D00UL)
```

System Control Block Base Address

#### 5.20.1.46 SCnSCB

```
#define SCnSCB ((SCnSCB_Type *) SCS_BASE )
```

System control Register not in SCB

#### 5.20.1.47 SCS\_BASE

```
#define SCS_BASE (0xE000E000UL)
```

System Control Space Base Address

#### 5.20.1.48 SRAM\_BASE

```
#define SRAM_BASE ((uint32_t)0x20000000)
```

SRAM base address in the alias region

#### 5.20.1.49 SRAM\_BB\_BASE

```
#define SRAM_BB_BASE ((uint32_t)0x22000000)
```

SRAM base address in the bit-band region

#### 5.20.1.50 SysTick

```
#define SysTick ((SysTick_Type *) SysTick_BASE )
```

SysTick configuration struct

#### 5.20.1.51 SysTick\_BASE

```
#define SysTick_BASE (SCS_BASE + 0x0010UL)
```

SysTick Base Address

#### 5.20.1.52 TIM1

```
#define TIM1 ((TIM_TypeDef *) TIM1_BASE)
```

#### 5.20.1.53 TIM1\_BASE

```
#define TIM1_BASE (APB2PERIPH_BASE + 0x2C00)
```

#### 5.20.1.54 TIM2

```
#define TIM2 ((TIM_TypeDef *) TIM2_BASE)
```

#### 5.20.1.55 TIM2\_BASE

```
#define TIM2_BASE (APB1PERIPH_BASE + 0x0000)
```

#### 5.20.1.56 TIM3

```
#define TIM3 ((TIM_TypeDef *) TIM3_BASE)
```

#### 5.20.1.57 TIM3\_BASE

```
#define TIM3_BASE (APB1PERIPH_BASE + 0x0400)
```

#### 5.20.1.58 TIM4

```
#define TIM4 ((TIM_TypeDef *) TIM4_BASE)
```

#### 5.20.1.59 TIM4\_BASE

```
#define TIM4_BASE (APB1PERIPH_BASE + 0x0800)
```

#### 5.20.1.60 TPI

```
#define TPI ((TPI_Type *) TPI_BASE )
```

TPI configuration struct

#### 5.20.1.61 TPI\_BASE

```
#define TPI_BASE (0xE0040000UL)
```

TPI Base Address

#### 5.20.1.62 USART1

```
#define USART1 ((USART_TypeDef *) USART1_BASE)
```

#### 5.20.1.63 USART1\_BASE

```
#define USART1_BASE (APB2PERIPH_BASE + 0x3800)
```

#### 5.20.1.64 USART2

```
#define USART2 ((USART_TypeDef *) USART2_BASE)
```

#### 5.20.1.65 USART2\_BASE

```
#define USART2_BASE (APB1PERIPH_BASE + 0x4400)
```

### 5.20.1.66 USART3

```
#define USART3 ((USART_TypeDef *) USART3_BASE)
```

### 5.20.1.67 USART3\_BASE

```
#define USART3_BASE (APB1PERIPH_BASE + 0x4800)
```

## 5.20.2 Typedef Documentation

### 5.20.2.1 IRQn\_Type

```
typedef enum IRQn IRQn_Type
```

## 5.20.3 Enumeration Type Documentation

### 5.20.3.1 IRQn

```
enum IRQn
```

#### Enumerator

NonMaskableInt_IRQn	2 Non Maskable Interrupt
MemoryManagement_IRQn	4 Cortex-M3 Memory Management Interrupt
BusFault_IRQn	5 Cortex-M3 Bus Fault Interrupt
UsageFault_IRQn	6 Cortex-M3 Usage Fault Interrupt
SVCall_IRQn	11 Cortex-M3 SV Call Interrupt
DebugMonitor_IRQn	12 Cortex-M3 Debug Monitor Interrupt
PendSV_IRQn	14 Cortex-M3 Pend SV Interrupt
SysTick_IRQn	15 Cortex-M3 System Tick Interrupt
WWDG_IRQn	Window WatchDog Interrupt
PVD_IRQn	PVD through EXTI Line detection Interrupt

## Enumerator

TAMPER_IRQn	Tamper Interrupt
RTC_IRQn	RTC global Interrupt
FLASH_IRQn	FLASH global Interrupt
RCC_IRQn	RCC global Interrupt
EXTI0_IRQn	EXTI Line0 Interrupt
EXTI1_IRQn	EXTI Line1 Interrupt
EXTI2_IRQn	EXTI Line2 Interrupt
EXTI3_IRQn	EXTI Line3 Interrupt
EXTI4_IRQn	EXTI Line4 Interrupt
DMA1_Channel1_IRQn	DMA1 Channel 1 global Interrupt
DMA1_Channel2_IRQn	DMA1 Channel 2 global Interrupt
DMA1_Channel3_IRQn	DMA1 Channel 3 global Interrupt
DMA1_Channel4_IRQn	DMA1 Channel 4 global Interrupt
DMA1_Channel5_IRQn	DMA1 Channel 5 global Interrupt
DMA1_Channel6_IRQn	DMA1 Channel 6 global Interrupt
DMA1_Channel7_IRQn	DMA1 Channel 7 global Interrupt
ADC1_2_IRQn	ADC1 and ADC2 global Interrupt
USB_HP_CAN1_TX_IRQn	USB Device High Priority or CAN1 TX Interrupts
USB_LP_CAN1_RX0_IRQn	USB Device Low Priority or CAN1 RX0 Interrupts
CAN1_RX1_IRQn	CAN1 RX1 Interrupt
CAN1_SCE_IRQn	CAN1 SCE Interrupt
EXTI9_5_IRQn	External Line[9:5] Interrupts
TIM1_BRK_IRQn	TIM1 Break Interrupt
TIM1_UP_IRQn	TIM1 Update Interrupt
TIM1_TRG_COM_IRQn	TIM1 Trigger and Commutation Interrupt
TIM1_CC_IRQn	TIM1 Capture Compare Interrupt
TIM2_IRQn	TIM2 global Interrupt

## Enumerator

TIM3_IRQn	TIM3 global Interrupt
TIM4_IRQn	TIM4 global Interrupt
I2C1_EV_IRQn	I2C1 Event Interrupt
I2C1_ER_IRQn	I2C1 Error Interrupt
I2C2_EV_IRQn	I2C2 Event Interrupt
I2C2_ER_IRQn	I2C2 Error Interrupt
SPI1_IRQn	SPI1 global Interrupt
SPI2_IRQn	SPI2 global Interrupt
USART1_IRQn	USART1 global Interrupt
USART2_IRQn	USART2 global Interrupt
USART3_IRQn	USART3 global Interrupt
EXTI15_10_IRQn	External Line[15:10] Interrupts
RTCArm_alarm_IRQn	RTC Alarm through EXTI Line Interrupt
USBWakeUp_IRQn	USB Device WakeUp from suspend through EXTI Line Interrupt

## 5.21 C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/MOTOR.cpp File Reference

```
#include "MOTOR.h"
```

## 5.22 C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/MOTOR.h File Reference

```
#include "PWM.h"
#include "ENCODER.h"
```

## Classes

- class [Motor](#)

## Macros

- `#define rpmToRads 2*pi/60`
- `#define rpmToV rpmToRads*r`
- `#define radsToRpm 60/(2*pi)`
- `#define r 0.0161`
- `#define desired_size 3`
- `#define k (desired_size-1)`
- `#define K 455.5`
- `#define P 12.97`
- `#define ts 0.15`
- `#define KP 4/(K*ts)`
- `#define KI P*KP`
- `#define Ts Time_between_int/1000000`

## Enumerations

- `enum MOTOR_ENUM {  
 Motor_1, Motor_2, Motor_3, Motor_4,  
 Number_of_Motor }`

### 5.22.1 Macro Definition Documentation

#### 5.22.1.1 desired\_size

```
#define desired_size 3
```

#### 5.22.1.2 k

```
#define k (desired_size-1)
```

#### 5.22.1.3 K

```
#define K 455.5
```

#### 5.22.1.4 KI

```
#define KI P*KP
```

#### 5.22.1.5 KP

```
#define KP 4/(K*ts)
```

#### 5.22.1.6 P

```
#define P 12.97
```

#### 5.22.1.7 r

```
#define r 0.0161
```

#### 5.22.1.8 radsToRpm

```
#define radsToRpm 60/(2*pi)
```

#### 5.22.1.9 rpmToRads

```
#define rpmToRads 2*pi/60
```

#### 5.22.1.10 rpmToV

```
#define rpmToV rpmToRads*r
```

#### 5.22.1.11 ts

```
#define ts 0.15
```

#### 5.22.1.12 Ts

```
#define Ts Time_between_int/1000000
```

### 5.22.2 Enumeration Type Documentation

#### 5.22.2.1 MOTOR\_ENUM

```
enum MOTOR_ENUM
```



## Enumerator

Motor_1	
Motor_2	
Motor_3	
Motor_4	
Number_of_Motor	

## 5.23 C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/PWM.cpp File Reference

```
#include "PWM.h"
```

## 5.24 C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/PWM.h File Reference

```
#include "GPIO.h"
#include "TIMER.h"
```

### Classes

- class [PWM](#)

### Macros

- #define [Timer\\_Frequency](#) 36000000
- #define [PWM\\_Frequency](#) 10000
- #define [AutoReloadPWM](#) (([Timer\\_Frequency](#)/[PWM\\_Frequency](#)) - 1)
- #define [Max\\_PWM](#) [AutoReloadPWM](#)

### 5.24.1 Macro Definition Documentation

#### 5.24.1.1 AutoReloadPWM

```
#define AutoReloadPWM ((Timer\_Frequency/PWM\_Frequency) - 1)
```

#### 5.24.1.2 Max\_PWM

```
#define Max_PWM AutoReloadPWM
```

#### 5.24.1.3 PWM\_Frequency

```
#define PWM_Frequency 10000
```

#### 5.24.1.4 Timer\_Frequency

```
#define Timer_Frequency 36000000
```

### 5.25 C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/REFLECTANCE\_SENSOR.cpp File Reference

```
#include "REFLECTANCE_SENSOR.h"
```

### 5.26 C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/REFLECTANCE\_SENSOR.h File Reference

```
#include "ADC.h"
```

## Classes

- class [Reflectance\\_Sensor](#)

### 5.27 C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/SERIAL\_COMMUNICATION.cpp File Reference

```
#include "SERIAL_COMMUNICATION.h"
```

## 5.28 C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/SERIAL\_COMMUNICATION.h File Reference

```
#include "FOLLOWING_CONTROLLER.h"
#include "USART.h"
```

### Classes

- class [Communication](#)

### Macros

- #define [samplingTime](#) 7
- #define [bitsToSend](#) ([samplingTime](#)\*1000000)/([Time\\_between\\_int](#))
- #define [sendingTime](#) [bitsToSend](#)\*[Time\\_between\\_int](#)/1000000
- #define [robotStop](#) 0
- #define [robotRun](#) 1
- #define [sendV\\_W](#) 2
- #define [sendV\\_Wcontrol](#) 3
- #define [sendV\\_W\\_and\\_V\\_Wcontrol](#) 4
- #define [sendLineReading](#) 5
- #define [sendMotorsSpeed](#) 6
- #define [sendMotorsSpeedAndControl](#) 7
- #define [sendMotorsSpeedControlAndErrors](#) 8
- #define [sendEverything](#) 9
- #define [sendPositionAndOrientation](#) 10
- #define [sendPWMandEncoderDataRight](#) 11
- #define [sendMotorDdata](#) 12

### 5.28.1 Macro Definition Documentation

#### 5.28.1.1 bitsToSend

```
#define bitsToSend (samplingTime*1000000)/(Time_between_int)
```

#### 5.28.1.2 robotRun

```
#define robotRun 1
```

**5.28.1.3 robotStop**

```
#define robotStop 0
```

**5.28.1.4 samplingTime**

```
#define samplingTime 7
```

**5.28.1.5 sendEverything**

```
#define sendEverything 9
```

**5.28.1.6 sendingTime**

```
#define sendingTime bitsToSend*Time_between_int/1000000
```

**5.28.1.7 sendLineReading**

```
#define sendLineReading 5
```

**5.28.1.8 sendMotorDdata**

```
#define sendMotorDdata 12
```

**5.28.1.9 sendMotorsSpeed**

```
#define sendMotorsSpeed 6
```

**5.28.1.10 sendMotorsSpeedAndControl**

```
#define sendMotorsSpeedAndControl 7
```

#### 5.28.1.11 sendMotorsSpeedControlAndErrors

```
#define sendMotorsSpeedControlAndErrors 8
```

#### 5.28.1.12 sendPositionAndOrientation

```
#define sendPositionAndOrientation 10
```

#### 5.28.1.13 sendPWMandEncoderDataRight

```
#define sendPWMandEncoderDataRight 11
```

#### 5.28.1.14 sendV\_W

```
#define sendV_W 2
```

#### 5.28.1.15 sendV\_W\_and\_V\_Wcontrol

```
#define sendV_W_and_V_Wcontrol 4
```

#### 5.28.1.16 sendV\_Wcontrol

```
#define sendV_Wcontrol 3
```

## 5.29 C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/SysClock.cpp File Reference

```
#include "SysClock.h"
```

## 5.30 C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/SysClock.h File Reference

```
#include "Micro.h"
#include "GPIO.h"
```

### Classes

- class [SysClock](#)

### Enumerations

- enum [SysTickBaseTimeEnum](#) {  
    [BASE\\_100ms](#) = (7200000-1), [BASE\\_10ms](#) = (720000 -1), [BASE\\_1ms](#) = (72000 -1), [BASE\\_100us](#) = (7200-1),  
    [BASE\\_10us](#) = (720 -1), [BASE\\_1us](#) = (72 -1) }

#### 5.30.1 Enumeration Type Documentation

##### 5.30.1.1 SysTickBaseTimeEnum

```
enum SysTickBaseTimeEnum
```

##### Enumerator

<a href="#">BASE_100ms</a>	
<a href="#">BASE_10ms</a>	
<a href="#">BASE_1ms</a>	
<a href="#">BASE_100us</a>	
<a href="#">BASE_10us</a>	
<a href="#">BASE_1us</a>	

## 5.31 C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/testes.cpp File Reference

## 5.32 C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/TIMER.cpp File Reference

```
#include "TIMER.h"
```

## 5.33 C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/TIMER.h File Reference

```
#include "GPIO.h"
#include "Micro.h"
```

### Classes

- class [Timer](#)

### Macros

- #define [Prescale](#) 71
- #define [Time\\_between\\_int](#) 1000
- #define [Time\\_between\\_int\\_milis](#) [Time\\_between\\_int](#)/1000
- #define [AutoReload\\_Counter](#) ([Time\\_between\\_int](#)-1)

### Enumerations

- enum [TIM\\_MODE](#) { [COUNTER](#) = 0, [PWM\\_MODE](#), [QUADRATURE\\_ENCODER\\_MODE](#) }
- enum [TIM\\_CHANNELS](#) { [TIM\\_CH1](#) = 0, [TIM\\_CH2](#), [TIM\\_CH3](#), [TIM\\_CH4](#) }
- enum [TIM\\_REMAP](#) { [NO\\_REMAP](#) = 0, [PARTIAL\\_REMAP1](#), [PARTIAL\\_REMAP2](#), [FULL\\_REMAP](#) }

### 5.33.1 Macro Definition Documentation

#### 5.33.1.1 AutoReload\_Counter

```
#define AutoReload_Counter (Time\_between\_int-1)
```

### 5.33.1.2 Prescale

```
#define Prescale 71
```

### 5.33.1.3 Time\_between\_int

```
#define Time_between_int 1000
```

### 5.33.1.4 Time\_between\_int\_milis

```
#define Time_between_int_milis Time\_between\_int/1000
```

## 5.33.2 Enumeration Type Documentation

### 5.33.2.1 TIM\_CHANNELS

```
enum TIM\_CHANNELS
```

Enumerator

TIM_CH1	
TIM_CH2	
TIM_CH3	
TIM_CH4	

### 5.33.2.2 TIM\_MODE

```
enum TIM\_MODE
```

Enumerator

COUNTER	
PWM_MODE	
QUADRATURE_ENCODER_MODE	



### 5.33.2.3 TIM\_REMAP

enum [TIM\\_REMAP](#)

Enumerator

NO_REMAP	
PARTIAL_REMAP1	
PARTIAL_REMAP2	
FULL_REMAP	

## 5.34 C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/USART.cpp File Reference

```
#include "USART.h"
```

### Functions

- void [sendUInt32](#) (uint32\_t \*ptUInt32)

### 5.34.1 Function Documentation

#### 5.34.1.1 sendUInt32()

```
void sendUInt32 (
    uint32_t * ptUInt32 )
```

## 5.35 C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/USART.h File Reference

```
#include "Micro.h"
#include "GPIO.h"
```

### Classes

- class [USART](#)

## Enumerations

- enum [BD\\_ENUM](#) {  
    [BD\\_9600](#), [BD\\_38400](#), [BD\\_57600](#), [BD\\_115200](#),  
    [BD\\_230769](#), [BD\\_250000](#), [BD\\_1382400](#), [BD\\_1000000](#),  
    [BD\\_2250000](#) }

### 5.35.1 Enumeration Type Documentation

#### 5.35.1.1 [BD\\_ENUM](#)

enum [BD\\_ENUM](#)

##### Enumerator

<a href="#">BD_9600</a>	
<a href="#">BD_38400</a>	
<a href="#">BD_57600</a>	
<a href="#">BD_115200</a>	
<a href="#">BD_230769</a>	
<a href="#">BD_250000</a>	
<a href="#">BD_1382400</a>	
<a href="#">BD_1000000</a>	
<a href="#">BD_2250000</a>	

# Index

ACR  
    FLASH\_TypeDef, [20](#)

ADC, [7](#)  
    ADC, [7](#)  
    analogRead, [8](#)  
    ConfigADCPin, [8](#)  
    GetADCChannel, [8](#)  
    SetADCChannel, [8](#)

ADC.h  
    ADC\_CH0, [70](#)  
    ADC\_CH1, [70](#)  
    ADC\_CH2, [70](#)  
    ADC\_CH3, [70](#)  
    ADC\_CH4, [70](#)  
    ADC\_CH5, [70](#)  
    ADC\_CH6, [70](#)  
    ADC\_CH7, [70](#)  
    ADC\_CH8, [70](#)  
    ADC\_CH9, [70](#)  
    ADC\_CHANNELS, [69](#)  
    ADC\_CONVERSION\_MODES, [70](#)  
    CONTINUOUS\_CONVERSION, [70](#)  
    SINGLE\_CONVERSION, [70](#)

ADC1  
    Micro.h, [89](#)

ADC1\_2\_IRQn  
    Micro.h, [99](#)

ADC1\_BASE  
    Micro.h, [89](#)

ADC2  
    Micro.h, [90](#)

ADC2\_BASE  
    Micro.h, [90](#)

ADC\_CH0  
    ADC.h, [70](#)

ADC\_CH1  
    ADC.h, [70](#)

ADC\_CH2  
    ADC.h, [70](#)

ADC\_CH3  
    ADC.h, [70](#)

ADC\_CH4  
    ADC.h, [70](#)

ADC\_CH5  
    ADC.h, [70](#)

ADC\_CH6  
    ADC.h, [70](#)

ADC\_CH7  
    ADC.h, [70](#)

ADC\_CH8  
    ADC.h, [70](#)

ADC\_CH9  
    ADC.h, [70](#)

ADC\_CHANNELS  
    ADC.h, [69](#)

ADC\_CONVERSION\_MODES  
    ADC.h, [70](#)

ADC\_TypeDef, [8](#)  
    CR1, [9](#)  
    CR2, [9](#)  
    DR, [9](#)  
    HTR, [9](#)  
    JDR1, [9](#)  
    JDR2, [10](#)  
    JDR3, [10](#)  
    JDR4, [10](#)  
    JOFR1, [10](#)  
    JOFR2, [10](#)  
    JOFR3, [10](#)  
    JOFR4, [10](#)  
    JSQR, [10](#)  
    LTR, [11](#)  
    SMPR1, [11](#)  
    SMPR2, [11](#)  
    SQR1, [11](#)  
    SQR2, [11](#)  
    SQR3, [11](#)  
    SR, [11](#)

AF\_OUTPUT\_OPEN\_DRAIN\_10MHZ  
    GPIO.h, [78](#)

AF\_OUTPUT\_OPEN\_DRAIN\_2MHZ  
    GPIO.h, [78](#)

AF\_OUTPUT\_OPEN\_DRAIN\_50MHZ  
    GPIO.h, [78](#)

AF\_OUTPUT\_PUSH\_PULL\_10MHZ  
    GPIO.h, [78](#)

AF\_OUTPUT\_PUSH\_PULL\_2MHZ  
    GPIO.h, [78](#)

AF\_OUTPUT\_PUSH\_PULL\_50MHZ  
    GPIO.h, [78](#)

AFIO  
    Micro.h, [90](#)

AFIO\_BASE  
    Micro.h, [90](#)

AFIO\_TypeDef, [12](#)  
    EVCr, [12](#)  
    EXTICR, [12](#)  
    MAPR, [12](#)

- MAPR2, [12](#)
- RESERVED0, [13](#)
- AHBENR
  - RCC\_TypeDef, [43](#)
- AHBPERIPH\_BASE
  - Micro.h, [90](#)
- ALRH
  - RTC\_TypeDef, [47](#)
- ALRL
  - RTC\_TypeDef, [47](#)
- analogRead
  - ADC, [8](#)
- angle
  - Kinematic, [31](#)
- ANGLE\_LINE\_SENSOR\_H, [13](#)
  - readAngle, [13](#)
  - updateLineAngle, [13](#)
- angleCorrection
  - KINEMATIC\_CONTROL.h, [80](#)
- angleFilterOrder
  - KINEMATIC\_CONTROL.h, [80](#)
- APB1ENR
  - RCC\_TypeDef, [44](#)
- APB1PERIPH\_BASE
  - Micro.h, [90](#)
- APB1RSTR
  - RCC\_TypeDef, [44](#)
- APB2ENR
  - RCC\_TypeDef, [44](#)
- APB2PERIPH\_BASE
  - Micro.h, [90](#)
- APB2RSTR
  - RCC\_TypeDef, [44](#)
- AR
  - FLASH\_TypeDef, [20](#)
- ARR
  - TIM\_TypeDef, [53](#)
- AutoReload\_Ticks
  - ENCODER.h, [71](#)
- AutoReload\_Counter
  - TIMER.h, [109](#)
- AutoReloadPWM
  - PWM.h, [103](#)
- Available
  - USART, [64](#)
- backward
  - ENCODER.h, [72](#)
- BASE\_100ms
  - SysClock.h, [108](#)
- BASE\_100us
  - SysClock.h, [108](#)
- BASE\_10ms
  - SysClock.h, [108](#)
- BASE\_10us
  - SysClock.h, [108](#)
- BASE\_1ms
  - SysClock.h, [108](#)
- BASE\_1us
  - SysClock.h, [108](#)
- BD\_1000000
  - USART.h, [112](#)
- BD\_115200
  - USART.h, [112](#)
- BD\_1382400
  - USART.h, [112](#)
- BD\_2250000
  - USART.h, [112](#)
- BD\_230769
  - USART.h, [112](#)
- BD\_250000
  - USART.h, [112](#)
- BD\_38400
  - USART.h, [112](#)
- BD\_57600
  - USART.h, [112](#)
- BD\_9600
  - USART.h, [112](#)
- BD\_ENUM
  - USART.h, [112](#)
- BDCR
  - RCC\_TypeDef, [44](#)
- BDTR
  - TIM\_TypeDef, [54](#)
- bitsToSend
  - SERIAL\_COMMUNICATION.h, [105](#)
- BRR
  - GPIO\_TypeDef, [26](#)
  - USART\_TypeDef, [66](#)
- BSRR
  - GPIO\_TypeDef, [26](#)
- BusFault\_IRQn
  - Micro.h, [98](#)
- C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/AD
  - [69](#)
- C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/AD
  - [69](#)
- C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/AN
  - [70](#)
- C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/AN
  - [70](#)
- C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/EN
  - [70](#)
- C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/EN
  - [71](#)
- C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/FO
  - [73](#)
- C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/FO
  - [73](#)
- C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/GP
  - [76](#)
- C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/GP
  - [76](#)
- C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/KIN
  - [79](#)
- C:/Users/bruno/Documents/LineFollower/Project\_uVision/LineFollower/KIN
  - [79](#)



- Controller, [16](#)
- CoreDebug
  - Micro.h, [90](#)
- CoreDebug\_BASE
  - Micro.h, [91](#)
- COUNTER
  - TIMER.h, [110](#)
- counter
  - malha\_fechada.cpp, [85](#)
- CR
  - FLASH\_TypeDef, [21](#)
  - PWR\_TypeDef, [43](#)
  - RCC\_TypeDef, [44](#)
- CR1
  - ADC\_TypeDef, [9](#)
  - TIM\_TypeDef, [55](#)
  - USART\_TypeDef, [66](#)
- CR2
  - ADC\_TypeDef, [9](#)
  - TIM\_TypeDef, [55](#)
  - USART\_TypeDef, [66](#)
- CR3
  - USART\_TypeDef, [66](#)
- CRH
  - GPIO\_TypeDef, [27](#)
  - RTC\_TypeDef, [47](#)
- CRL
  - GPIO\_TypeDef, [27](#)
  - RTC\_TypeDef, [48](#)
- CSR
  - PWR\_TypeDef, [43](#)
  - RCC\_TypeDef, [45](#)
- CTRL
  - SysTick\_Type, [52](#)
- DCR
  - TIM\_TypeDef, [55](#)
- DebugMonitor\_IRQn
  - Micro.h, [98](#)
- delay
  - Timer, [60](#)
- delta\_time
  - malha\_fechada.cpp, [85](#)
- deltaDistance
  - Kinematic, [31](#)
- desired\_size
  - MOTOR.h, [101](#)
- DIER
  - TIM\_TypeDef, [55](#)
- digitalRead
  - GPIO, [23](#)
- digitalWrite
  - GPIO, [24](#)
- distance
  - Kinematic, [31](#)
- distanceCorrection
  - KINEMATIC\_CONTROL.h, [80](#)
- DIVH
  - RTC\_TypeDef, [48](#)
- DIVL
  - RTC\_TypeDef, [48](#)
- DMA1\_Channel1\_IRQn
  - Micro.h, [99](#)
- DMA1\_Channel2\_IRQn
  - Micro.h, [99](#)
- DMA1\_Channel3\_IRQn
  - Micro.h, [99](#)
- DMA1\_Channel4\_IRQn
  - Micro.h, [99](#)
- DMA1\_Channel5\_IRQn
  - Micro.h, [99](#)
- DMA1\_Channel6\_IRQn
  - Micro.h, [99](#)
- DMA1\_Channel7\_IRQn
  - Micro.h, [99](#)
- DMAR
  - TIM\_TypeDef, [55](#)
- DR
  - ADC\_TypeDef, [9](#)
  - USART\_TypeDef, [67](#)
- DWT
  - Micro.h, [91](#)
- DWT\_BASE
  - Micro.h, [91](#)
- E
  - Motor, [37](#)
- EGR
  - TIM\_TypeDef, [55](#)
- Encoder, [17](#)
  - Encoder, [18](#)
  - Encoder\_Handler, [18](#)
  - Encoder\_Handler\_by\_Time, [18](#)
  - Encoder\_Initialize, [18](#)
  - getDeltaTicks, [18](#)
  - getDeltaTime, [19](#)
  - getDirection, [19](#)
  - getLastSpeed, [19](#)
  - getLastTicksTime, [19](#)
  - getNotFilteredSpeed, [19](#)
  - getSpeed, [19](#)
  - getTeta, [19](#)
  - getTicks, [19](#)
  - getTicksTime, [20](#)
  - reset, [20](#)
- encoder
  - Motor, [37](#)
- ENCODER.h
  - AutoReload\_Ticks, [71](#)
  - backward, [72](#)
  - ENCODER\_DIRECTION, [72](#)
  - ENCODER\_ENUM, [72](#)
  - Encoder\_TIM1, [73](#)
  - Encoder\_TIM2, [73](#)
  - Encoder\_TIM3, [73](#)
  - Encoder\_TIM4, [73](#)
  - encoderFilterOrder, [71](#)
  - forward, [72](#)

- Max\_delay\_Ticks\_Time, [71](#)
- NUMBER\_OF\_ENCODERS, [73](#)
- pi, [72](#)
- pulsePerRevolution, [72](#)
- Ticks\_till\_int, [72](#)
- ticksToRad, [72](#)
- ENCODER\_DIRECTION
  - ENCODER.h, [72](#)
- ENCODER\_ENUM
  - ENCODER.h, [72](#)
- Encoder\_Handler
  - Encoder, [18](#)
- Encoder\_Handler\_by\_Time
  - Encoder, [18](#)
- Encoder\_Initialize
  - Encoder, [18](#)
- Encoder\_TIM1
  - ENCODER.h, [73](#)
- Encoder\_TIM2
  - ENCODER.h, [73](#)
- Encoder\_TIM3
  - ENCODER.h, [73](#)
- Encoder\_TIM4
  - ENCODER.h, [73](#)
- encoderFilterOrder
  - ENCODER.h, [71](#)
- EVCR
  - AFIO\_TypeDef, [12](#)
- EXTI0\_IRQn
  - Micro.h, [99](#)
- EXTI15\_10\_IRQn
  - Micro.h, [100](#)
- EXTI1\_IRQn
  - Micro.h, [99](#)
- EXTI2\_IRQn
  - Micro.h, [99](#)
- EXTI3\_IRQn
  - Micro.h, [99](#)
- EXTI4\_IRQn
  - Micro.h, [99](#)
- EXTI9\_5\_IRQn
  - Micro.h, [99](#)
- EXTICR
  - AFIO\_TypeDef, [12](#)
- filteredAngle
  - Kinematic, [31](#)
- FLASH
  - Micro.h, [91](#)
- FLASH\_BASE
  - Micro.h, [91](#)
- FLASH\_IRQn
  - Micro.h, [99](#)
- FLASH\_R\_BASE
  - Micro.h, [91](#)
- FLASH\_TypeDef, [20](#)
  - ACR, [20](#)
  - AR, [20](#)
  - CR, [21](#)
- KEYR, [21](#)
- OBR, [21](#)
- OPTKEYR, [21](#)
- RESERVED, [21](#)
- SR, [21](#)
- WRPR, [21](#)
- FOLLOWING\_CONTROLER.h
  - K1, [74](#)
  - K1a, [74](#)
  - K2, [74](#)
  - K2a, [74](#)
  - K3, [74](#)
  - K4, [74](#)
  - K4k, [74](#)
  - Kteta, [74](#)
  - Ktetak, [75](#)
  - Kx, [75](#)
  - Kxk, [75](#)
  - Ky, [75](#)
  - Kyk, [75](#)
  - v\_ref\_ini, [75](#)
  - w\_ref\_ini, [75](#)
- forward
  - ENCODER.h, [72](#)
- FSMC\_R\_BASE
  - Micro.h, [91](#)
- FULL\_REMAP
  - TIMER.h, [111](#)
- Get\_Speed
  - Motor, [36](#)
- GetADCChannel
  - ADC, [8](#)
- getDeltaDistance
  - Motor, [36](#)
- getDeltaTicks
  - Encoder, [18](#)
- getDeltaTime
  - Encoder, [19](#)
- getDirection
  - Encoder, [19](#)
- getDistance
  - Motor, [36](#)
- getE
  - Motor, [36](#)
- GetGPIOMode
  - GPIO, [24](#)
- GetGPIOPinNumber
  - GPIO, [24](#)
- GetGPIOPort
  - GPIO, [24](#)
- GetGPIOPuPd
  - GPIO, [24](#)
- GetGPIOState
  - GPIO, [24](#)
- GetIOPin
  - GPIO, [24](#)
- getLastSpeed
  - Encoder, [19](#)

- getLastTicksTime
  - Encoder, [19](#)
- getLineAngle
  - Kinematic, [29](#)
- getLineAngleNotFiltered
  - Kinematic, [29](#)
- getLinePosition
  - Kinematic, [29](#)
- getNotFilteredSpeed
  - Encoder, [19](#)
- getSpeed
  - Encoder, [19](#)
- getSpeedRadS
  - Motor, [36](#)
- getTeta
  - Encoder, [19](#)
  - Kinematic, [29](#)
  - Motor, [36](#)
- getTicks
  - Encoder, [19](#)
- getTicksTime
  - Encoder, [20](#)
- GetTim
  - Timer, [60](#)
- GetTIMChannel
  - Timer, [61](#)
- GetTime\_milise
  - Timer, [61](#)
- GetTime\_usec
  - Timer, [61](#)
- GetTIMMode
  - Timer, [61](#)
- GetTIMRemap
  - Timer, [61](#)
- getU
  - Motor, [36](#)
- getV
  - Kinematic, [30](#)
- getVcontrol
  - Controller, [16](#)
- getW
  - Kinematic, [30](#)
- getWcontrol
  - Controller, [16](#)
- getWritedPWM
  - PWM, [42](#)
- getX
  - Kinematic, [30](#)
- getY
  - Kinematic, [30](#)
- GP\_OUTPUT\_OPEN\_DRAIN\_10MHZ
  - GPIO.h, [78](#)
- GP\_OUTPUT\_OPEN\_DRAIN\_2MHZ
  - GPIO.h, [78](#)
- GP\_OUTPUT\_OPEN\_DRAIN\_50MHZ
  - GPIO.h, [78](#)
- GP\_OUTPUT\_PUSH\_PULL\_10MHZ
  - GPIO.h, [78](#)
- GP\_OUTPUT\_PUSH\_PULL\_2MHZ
  - GPIO.h, [78](#)
- GP\_OUTPUT\_PUSH\_PULL\_50MHZ
  - GPIO.h, [78](#)
- GPIO, [22](#)
  - Config\_PU\_PD, [23](#)
  - ConfigGPIOPin, [23](#)
  - digitalRead, [23](#)
  - digitalWrite, [24](#)
  - GetGPIOMode, [24](#)
  - GetGPIOPinNumber, [24](#)
  - GetGPIOPort, [24](#)
  - GetGPIOPuPd, [24](#)
  - GetGPIOState, [24](#)
  - GetIOPin, [24](#)
  - GPIO, [22](#), [23](#)
  - SetGPIOMode, [25](#)
  - SetGPIOPortPin, [25](#)
  - SetIOPin, [25](#)
  - togglePin, [25](#)
- GPIO.h
  - AF\_OUTPUT\_OPEN\_DRAIN\_10MHZ, [78](#)
  - AF\_OUTPUT\_OPEN\_DRAIN\_2MHZ, [78](#)
  - AF\_OUTPUT\_OPEN\_DRAIN\_50MHZ, [78](#)
  - AF\_OUTPUT\_PUSH\_PULL\_10MHZ, [78](#)
  - AF\_OUTPUT\_PUSH\_PULL\_2MHZ, [78](#)
  - AF\_OUTPUT\_PUSH\_PULL\_50MHZ, [78](#)
  - GP\_OUTPUT\_OPEN\_DRAIN\_10MHZ, [78](#)
  - GP\_OUTPUT\_OPEN\_DRAIN\_2MHZ, [78](#)
  - GP\_OUTPUT\_OPEN\_DRAIN\_50MHZ, [78](#)
  - GP\_OUTPUT\_PUSH\_PULL\_10MHZ, [78](#)
  - GP\_OUTPUT\_PUSH\_PULL\_2MHZ, [78](#)
  - GP\_OUTPUT\_PUSH\_PULL\_50MHZ, [78](#)
  - GPIO\_IO\_ENUM, [77](#)
  - GPIO\_MODES, [78](#)
  - HIGH, [77](#)
  - INPUT\_ANALOG, [78](#)
  - INPUT\_FLOATING, [78](#)
  - INPUT\_PULL\_UP\_DOWN, [78](#)
  - LOW, [77](#)
  - NUM\_OF\_IOs, [78](#)
  - NUMBER\_OF\_PINS, [79](#)
  - PA0, [77](#)
  - PA1, [77](#)
  - PA10, [77](#)
  - PA11, [77](#)
  - PA12, [77](#)
  - PA15, [77](#)
  - PA2, [77](#)
  - PA3, [77](#)
  - PA4, [77](#)
  - PA5, [77](#)
  - PA6, [77](#)
  - PA7, [77](#)
  - PA8, [77](#)
  - PA9, [77](#)
  - PB0, [77](#)
  - PB1, [77](#)



- PB10, [77](#)
- PB11, [78](#)
- PB12, [78](#)
- PB13, [78](#)
- PB14, [78](#)
- PB15, [78](#)
- PB3, [77](#)
- PB4, [77](#)
- PB5, [77](#)
- PB6, [77](#)
- PB7, [77](#)
- PB8, [77](#)
- PB9, [77](#)
- PC13, [78](#)
- PC14, [78](#)
- PC15, [78](#)
- PIN0, [78](#)
- PIN1, [78](#)
- PIN10, [79](#)
- PIN11, [79](#)
- PIN12, [79](#)
- PIN13, [79](#)
- PIN14, [79](#)
- PIN15, [79](#)
- PIN2, [78](#)
- PIN3, [78](#)
- PIN4, [78](#)
- PIN5, [79](#)
- PIN6, [79](#)
- PIN7, [79](#)
- PIN8, [79](#)
- PIN9, [79](#)
- PIN\_NUMBERS, [78](#)
- PU\_PD\_ENUM, [79](#)
- PULL\_DOWN, [79](#)
- PULL\_UP, [79](#)
- GPIO\_IO\_ENUM
  - GPIO.h, [77](#)
- GPIO\_MODES
  - GPIO.h, [78](#)
- GPIO\_STRUCT, [25](#)
  - pinNumber, [26](#)
  - Port, [26](#)
- GPIO\_TypeDef, [26](#)
  - BRR, [26](#)
  - BSRR, [26](#)
  - CRH, [27](#)
  - CRL, [27](#)
  - IDR, [27](#)
  - LCKR, [27](#)
  - ODR, [27](#)
- GPIOA
  - Micro.h, [91](#)
- GPIOA\_BASE
  - Micro.h, [92](#)
- GPIOB
  - Micro.h, [92](#)
- GPIOB\_BASE
  - Micro.h, [92](#)
- GPIOC
  - Micro.h, [92](#)
- GPIOC\_BASE
  - Micro.h, [92](#)
- GPIOD
  - Micro.h, [92](#)
- GPIOD\_BASE
  - Micro.h, [92](#)
- GPIOE
  - Micro.h, [92](#)
- GPIOE\_BASE
  - Micro.h, [93](#)
- GPIOF
  - Micro.h, [93](#)
- GPIOF\_BASE
  - Micro.h, [93](#)
- GPIOG
  - Micro.h, [93](#)
- GPIOG\_BASE
  - Micro.h, [93](#)
- GTPR
  - USART\_TypeDef, [67](#)
- Handler
  - Motor, [37](#)
- handler
  - Kinematic, [30](#)
- HandlerByTime
  - Communication, [14](#)
  - Controller, [16](#)
- handlerByTime
  - Kinematic, [30](#)
- HIGH
  - GPIO.h, [77](#)
- HTR
  - ADC\_TypeDef, [9](#)
- I2C1\_ER\_IRQn
  - Micro.h, [100](#)
- I2C1\_EV\_IRQn
  - Micro.h, [100](#)
- I2C2\_ER\_IRQn
  - Micro.h, [100](#)
- I2C2\_EV\_IRQn
  - Micro.h, [100](#)
- IABR
  - NVIC\_Type, [39](#)
- ICER
  - NVIC\_Type, [39](#)
- ICPR
  - NVIC\_Type, [39](#)
- IDR
  - GPIO\_TypeDef, [27](#)
- IN1
  - Motor, [38](#)
- IN2
  - Motor, [38](#)
- INPUT\_ANALOG

- GPIO.h, 78
- INPUT\_FLOATING
  - GPIO.h, 78
- INPUT\_PULL\_UP\_DOWN
  - GPIO.h, 78
- integrationTime
  - KINEMATIC\_CONTROL.h, 80
- InterruptTime
  - Timer, 61
- IP
  - NVIC\_Type, 40
- IRQn
  - Micro.h, 98
- IRQn\_Type
  - Micro.h, 98
- ISER
  - NVIC\_Type, 40
- ISPR
  - NVIC\_Type, 40
- ITM
  - Micro.h, 93
- ITM\_BASE
  - Micro.h, 93
- JDR1
  - ADC\_TypeDef, 9
- JDR2
  - ADC\_TypeDef, 10
- JDR3
  - ADC\_TypeDef, 10
- JDR4
  - ADC\_TypeDef, 10
- JOFR1
  - ADC\_TypeDef, 10
- JOFR2
  - ADC\_TypeDef, 10
- JOFR3
  - ADC\_TypeDef, 10
- JOFR4
  - ADC\_TypeDef, 10
- JSQR
  - ADC\_TypeDef, 10
- K
  - MOTOR.h, 101
- k
  - MOTOR.h, 101
- K1
  - FOLLOWING\_CONTROLLER.h, 74
- K1a
  - FOLLOWING\_CONTROLLER.h, 74
- K2
  - FOLLOWING\_CONTROLLER.h, 74
- K2a
  - FOLLOWING\_CONTROLLER.h, 74
- K3
  - FOLLOWING\_CONTROLLER.h, 74
- K4
  - FOLLOWING\_CONTROLLER.h, 74
- K4k
  - FOLLOWING\_CONTROLLER.h, 74
- KDw
  - KINEMATIC\_CONTROL.h, 80
- KEYR
  - FLASH\_TypeDef, 21
- KI
  - MOTOR.h, 101
- Kinematic, 27
  - angle, 31
  - calibrateLineSensor, 29
  - calibrationFinished, 31
  - deltaDistance, 31
  - distance, 31
  - filteredAngle, 31
  - getLineAngle, 29
  - getLineAngleNotFiltered, 29
  - getLinePosition, 29
  - getTeta, 29
  - getV, 30
  - getW, 30
  - getX, 30
  - getY, 30
  - handler, 30
  - handlerByTime, 30
  - Kinematic, 28, 29
  - lastDistance, 31
  - lastLineSensorReading, 32
  - lineSensor, 32
  - lineSensorReading, 32
  - motorD, 32
  - motorE, 32
  - ptRobot, 32
  - reset, 30
  - setSpeed, 30
  - updateLineReading, 31
  - xPos, 32
  - yPos, 32
- KINEMATIC\_CONTROL.h
  - angleCorrection, 80
  - angleFilterOrder, 80
  - distanceCorrection, 80
  - integrationTime, 80
  - KDw, 80
  - KPw, 80
  - Kw, 81
  - L, 81
  - Pw, 81
  - ts\_w, 81
- KP
  - MOTOR.h, 101
- KPw
  - KINEMATIC\_CONTROL.h, 80
- Kteta
  - FOLLOWING\_CONTROLLER.h, 74
- Ktetak
  - FOLLOWING\_CONTROLLER.h, 75
- Kw

- KINEMATIC\_CONTROL.h, 81
- Kx
  - FOLLOWING\_CONTROLER.h, 75
- Kxk
  - FOLLOWING\_CONTROLER.h, 75
- Ky
  - FOLLOWING\_CONTROLER.h, 75
- Kyk
  - FOLLOWING\_CONTROLER.h, 75
- L
  - KINEMATIC\_CONTROL.h, 81
- lastDistance
  - Kinematic, 31
- lastLineSensorReading
  - Kinematic, 32
- LCKR
  - GPIO\_TypeDef, 27
- Line\_Sensor, 33
  - calibrate, 34
  - Line\_Sensor, 33
  - read, 34
  - Reflectance\_Sensor, 46
  - Sensors, 34
- LINE\_SENSOR.h
  - calib\_sensores, 82
  - maxDistance, 82
  - sensorFilterOrder, 82
- lineangle
  - malha\_fechada.cpp, 86
- lineSensor
  - Kinematic, 32
- lineSensorReading
  - Kinematic, 32
- linevalue
  - malha\_fechada.cpp, 86
- LOAD
  - SysTick\_Type, 52
- LOW
  - GPIO.h, 77
- LTR
  - ADC\_TypeDef, 11
- main
  - malha\_fechada.cpp, 84
- malha\_fechada.cpp
  - counter, 85
  - delta\_time, 85
  - lineangle, 86
  - linevalue, 86
  - main, 84
  - pwmtest, 86
  - RPM\_value, 86
  - steps, 84
  - stepTime, 84
  - t\_ini, 86
  - testTime, 84
  - TIM1\_UP\_IRQHandler, 85
  - TIM2\_IRQHandler, 85
  - TIM3\_IRQHandler, 85
  - TIM4\_IRQHandler, 85
  - time, 86
  - time\_test, 86
  - vtest, 86
  - wtest, 87
  - xtest, 87
  - ytest, 87
- MAPR
  - AFIO\_TypeDef, 12
- MAPR2
  - AFIO\_TypeDef, 12
- Max\_delay\_Ticks\_Time
  - ENCODER.h, 71
- Max\_PWM
  - PWM.h, 103
- maxDistance
  - LINE\_SENSOR.h, 82
- MCO
  - SysClock, 50
- MemoryManagement\_IRQn
  - Micro.h, 98
- Micro.h
  - ADC1, 89
  - ADC1\_2\_IRQn, 99
  - ADC1\_BASE, 89
  - ADC2, 90
  - ADC2\_BASE, 90
  - AFIO, 90
  - AFIO\_BASE, 90
  - AHBPERIPH\_BASE, 90
  - APB1PERIPH\_BASE, 90
  - APB2PERIPH\_BASE, 90
  - BusFault\_IRQn, 98
  - CAN1\_RX1\_IRQn, 99
  - CAN1\_SCE\_IRQn, 99
  - CoreDebug, 90
  - CoreDebug\_BASE, 91
  - DebugMonitor\_IRQn, 98
  - DMA1\_Channel1\_IRQn, 99
  - DMA1\_Channel2\_IRQn, 99
  - DMA1\_Channel3\_IRQn, 99
  - DMA1\_Channel4\_IRQn, 99
  - DMA1\_Channel5\_IRQn, 99
  - DMA1\_Channel6\_IRQn, 99
  - DMA1\_Channel7\_IRQn, 99
  - DWT, 91
  - DWT\_BASE, 91
  - EXTI0\_IRQn, 99
  - EXTI15\_10\_IRQn, 100
  - EXTI1\_IRQn, 99
  - EXTI2\_IRQn, 99
  - EXTI3\_IRQn, 99
  - EXTI4\_IRQn, 99
  - EXTI9\_5\_IRQn, 99
  - FLASH, 91
  - FLASH\_BASE, 91
  - FLASH\_IRQn, 99

FLASH\_R\_BASE, 91  
 FSMC\_R\_BASE, 91  
 GPIOA, 91  
 GPIOA\_BASE, 92  
 GPIOB, 92  
 GPIOB\_BASE, 92  
 GPIOC, 92  
 GPIOC\_BASE, 92  
 GPIOD, 92  
 GPIOD\_BASE, 92  
 GPIOE, 92  
 GPIOE\_BASE, 93  
 GPIOF, 93  
 GPIOF\_BASE, 93  
 GPIOG, 93  
 GPIOG\_BASE, 93  
 I2C1\_ER\_IRQn, 100  
 I2C1\_EV\_IRQn, 100  
 I2C2\_ER\_IRQn, 100  
 I2C2\_EV\_IRQn, 100  
 IRQn, 98  
 IRQn\_Type, 98  
 ITM, 93  
 ITM\_BASE, 93  
 MemoryManagement\_IRQn, 98  
 NonMaskableInt\_IRQn, 98  
 NVIC, 93  
 NVIC\_BASE, 94  
 PendSV\_IRQn, 98  
 PERIPH\_BASE, 94  
 PERIPH\_BB\_BASE, 94  
 PVD\_IRQn, 98  
 PWR, 94  
 PWR\_BASE, 94  
 RCC, 94  
 RCC\_BASE, 94  
 RCC\_IRQn, 99  
 RTC, 94  
 RTC\_BASE, 95  
 RTC\_IRQn, 99  
 RTCAlarm\_IRQn, 100  
 SCB, 95  
 SCB\_BASE, 95  
 SCnSCB, 95  
 SCS\_BASE, 95  
 SPI1\_IRQn, 100  
 SPI2\_IRQn, 100  
 SRAM\_BASE, 95  
 SRAM\_BB\_BASE, 95  
 SVCall\_IRQn, 98  
 SysTick, 95  
 SysTick\_BASE, 96  
 SysTick\_IRQn, 98  
 TAMPER\_IRQn, 99  
 TIM1, 96  
 TIM1\_BASE, 96  
 TIM1\_BRK\_IRQn, 99  
 TIM1\_CC\_IRQn, 99  
 TIM1\_TRG\_COM\_IRQn, 99  
 TIM1\_UP\_IRQn, 99  
 TIM2, 96  
 TIM2\_BASE, 96  
 TIM2\_IRQn, 99  
 TIM3, 96  
 TIM3\_BASE, 96  
 TIM3\_IRQn, 100  
 TIM4, 96  
 TIM4\_BASE, 97  
 TIM4\_IRQn, 100  
 TPI, 97  
 TPI\_BASE, 97  
 UsageFault\_IRQn, 98  
 USART1, 97  
 USART1\_BASE, 97  
 USART1\_IRQn, 100  
 USART2, 97  
 USART2\_BASE, 97  
 USART2\_IRQn, 100  
 USART3, 97  
 USART3\_BASE, 98  
 USART3\_IRQn, 100  
 USB\_HP\_CAN1\_TX\_IRQn, 99  
 USB\_LP\_CAN1\_RX0\_IRQn, 99  
 USBWakeUp\_IRQn, 100  
 WWDG\_IRQn, 98  
 Motor, 34  
   E, 37  
   encoder, 37  
   Get\_Speed, 36  
   getDeltaDistance, 36  
   getDistance, 36  
   getE, 36  
   getSpeedRadS, 36  
   getTeta, 36  
   getU, 36  
   Handler, 37  
   IN1, 38  
   IN2, 38  
   Motor, 35  
   Motor\_Handler\_by\_time, 37  
   Motor\_Initialiize, 37  
   Motor\_number, 38  
   motPtr, 38  
   pwmMotor, 38  
   reset, 37  
   Set\_Speed, 37  
   Speed\_Reference, 38  
   U, 38  
 MOTOR.h  
   desired\_size, 101  
   K, 101  
   k, 101  
   KI, 101  
   KP, 101  
   Motor\_1, 103  
   Motor\_2, 103

- Motor\_3, [103](#)
- Motor\_4, [103](#)
- MOTOR\_ENUM, [102](#)
- Number\_of\_Motor, [103](#)
- P, [102](#)
- r, [102](#)
- radsToRpm, [102](#)
- rpmToRads, [102](#)
- rpmToV, [102](#)
- Ts, [102](#)
- ts, [102](#)
- Motor\_1
  - MOTOR.h, [103](#)
- Motor\_2
  - MOTOR.h, [103](#)
- Motor\_3
  - MOTOR.h, [103](#)
- Motor\_4
  - MOTOR.h, [103](#)
- MOTOR\_ENUM
  - MOTOR.h, [102](#)
- Motor\_Handler\_by\_time
  - Motor, [37](#)
- Motor\_Initialize
  - Motor, [37](#)
- Motor\_number
  - Motor, [38](#)
- motorD
  - Kinematic, [32](#)
- motorE
  - Kinematic, [32](#)
- motPtr
  - Motor, [38](#)
- NO\_REMAP
  - TIMER.h, [111](#)
- NonMaskableInt\_IRQn
  - Micro.h, [98](#)
- NUM\_OF\_IOS
  - GPIO.h, [78](#)
- NUMBER\_OF\_ENCODERS
  - ENCODER.h, [73](#)
- Number\_of\_Motor
  - MOTOR.h, [103](#)
- NUMBER\_OF\_PINS
  - GPIO.h, [79](#)
- NVIC
  - Micro.h, [93](#)
- NVIC\_BASE
  - Micro.h, [94](#)
- NVIC\_Type, [39](#)
  - IABR, [39](#)
  - ICER, [39](#)
  - ICPR, [39](#)
  - IP, [40](#)
  - ISER, [40](#)
  - ISPR, [40](#)
  - RESERVED0, [40](#)
  - RESERVED2, [40](#)
  - RESERVED3, [40](#)
  - RESERVED4, [40](#)
  - RESERVED5, [40](#)
  - RSERVED1, [41](#)
  - STIR, [41](#)
- OBR
  - FLASH\_TypeDef, [21](#)
- ODR
  - GPIO\_TypeDef, [27](#)
- OPTKEYR
  - FLASH\_TypeDef, [21](#)
- P
  - MOTOR.h, [102](#)
- PA0
  - GPIO.h, [77](#)
- PA1
  - GPIO.h, [77](#)
- PA10
  - GPIO.h, [77](#)
- PA11
  - GPIO.h, [77](#)
- PA12
  - GPIO.h, [77](#)
- PA15
  - GPIO.h, [77](#)
- PA2
  - GPIO.h, [77](#)
- PA3
  - GPIO.h, [77](#)
- PA4
  - GPIO.h, [77](#)
- PA5
  - GPIO.h, [77](#)
- PA6
  - GPIO.h, [77](#)
- PA7
  - GPIO.h, [77](#)
- PA8
  - GPIO.h, [77](#)
- PA9
  - GPIO.h, [77](#)
- PARTIAL\_REMAP1
  - TIMER.h, [111](#)
- PARTIAL\_REMAP2
  - TIMER.h, [111](#)
- PB0
  - GPIO.h, [77](#)
- PB1
  - GPIO.h, [77](#)
- PB10
  - GPIO.h, [77](#)
- PB11
  - GPIO.h, [78](#)
- PB12
  - GPIO.h, [78](#)
- PB13
  - GPIO.h, [78](#)

- PB14
  - GPIO.h, [78](#)
- PB15
  - GPIO.h, [78](#)
- PB3
  - GPIO.h, [77](#)
- PB4
  - GPIO.h, [77](#)
- PB5
  - GPIO.h, [77](#)
- PB6
  - GPIO.h, [77](#)
- PB7
  - GPIO.h, [77](#)
- PB8
  - GPIO.h, [77](#)
- PB9
  - GPIO.h, [77](#)
- PC13
  - GPIO.h, [78](#)
- PC14
  - GPIO.h, [78](#)
- PC15
  - GPIO.h, [78](#)
- PendSV\_IRQn
  - Micro.h, [98](#)
- PERIPH\_BASE
  - Micro.h, [94](#)
- PERIPH\_BB\_BASE
  - Micro.h, [94](#)
- pi
  - ENCODER.h, [72](#)
- PIN0
  - GPIO.h, [78](#)
- PIN1
  - GPIO.h, [78](#)
- PIN10
  - GPIO.h, [79](#)
- PIN11
  - GPIO.h, [79](#)
- PIN12
  - GPIO.h, [79](#)
- PIN13
  - GPIO.h, [79](#)
- PIN14
  - GPIO.h, [79](#)
- PIN15
  - GPIO.h, [79](#)
- PIN2
  - GPIO.h, [78](#)
- PIN3
  - GPIO.h, [78](#)
- PIN4
  - GPIO.h, [78](#)
- PIN5
  - GPIO.h, [79](#)
- PIN6
  - GPIO.h, [79](#)
- PIN7
  - GPIO.h, [79](#)
- PIN8
  - GPIO.h, [79](#)
- PIN9
  - GPIO.h, [79](#)
- PIN\_NUMBERS
  - GPIO.h, [78](#)
- pinNumber
  - GPIO\_STRUCT, [26](#)
- Port
  - GPIO\_STRUCT, [26](#)
- Prescale
  - TIMER.h, [109](#)
- PRLH
  - RTC\_TypeDef, [48](#)
- PRLL
  - RTC\_TypeDef, [48](#)
- PSC
  - TIM\_TypeDef, [55](#)
- ptRobot
  - Kinematic, [32](#)
- PU\_PD\_ENUM
  - GPIO.h, [79](#)
- PULL\_DOWN
  - GPIO.h, [79](#)
- PULL\_UP
  - GPIO.h, [79](#)
- pulsePerRevolution
  - ENCODER.h, [72](#)
- PVD\_IRQn
  - Micro.h, [98](#)
- Pw
  - KINEMATIC\_CONTROL.h, [81](#)
- PWM, [41](#)
  - getWrtedPWM, [42](#)
  - PWM, [41](#), [42](#)
  - PWMWrite, [42](#)
- PWM.h
  - AutoReloadPWM, [103](#)
  - Max\_PWM, [103](#)
  - PWM\_Frequency, [104](#)
  - Timer\_Frequency, [104](#)
- PWM\_Frequency
  - PWM.h, [104](#)
- PWM\_MODE
  - TIMER.h, [110](#)
- pwmMotor
  - Motor, [38](#)
- pwmtest
  - malha\_fechada.cpp, [86](#)
- PWMWrite
  - PWM, [42](#)
- PWR
  - Micro.h, [94](#)
- PWR\_BASE
  - Micro.h, [94](#)
- PWR\_TypeDef, [42](#)

- CR, [43](#)
- CSR, [43](#)
- QUADRATURE\_ENCODER\_MODE
  - TIMER.h, [110](#)
- r
  - MOTOR.h, [102](#)
- radsToRpm
  - MOTOR.h, [102](#)
- RCC
  - Micro.h, [94](#)
- RCC\_BASE
  - Micro.h, [94](#)
- RCC\_IRQn
  - Micro.h, [99](#)
- RCC\_TypeDef, [43](#)
  - AHBENR, [43](#)
  - APB1ENR, [44](#)
  - APB1RSTR, [44](#)
  - APB2ENR, [44](#)
  - APB2RSTR, [44](#)
  - BDCR, [44](#)
  - CFGR, [44](#)
  - CIR, [44](#)
  - CR, [44](#)
  - CSR, [45](#)
- RCR
  - TIM\_TypeDef, [56](#)
- read
  - Line\_Sensor, [34](#)
- readAngle
  - ANGLE\_LINE\_SENSOR\_H, [13](#)
- Receive
  - USART, [65](#)
- Reflectance\_Read
  - Reflectance\_Sensor, [46](#)
- Reflectance\_Sensor, [45](#)
  - Calib\_Reflectance\_Sensor, [46](#)
  - Line\_Sensor, [46](#)
  - Reflectance\_Read, [46](#)
  - Reflectance\_Sensor, [45, 46](#)
- RESERVED
  - FLASH\_TypeDef, [21](#)
- RESERVED0
  - AFIO\_TypeDef, [13](#)
  - NVIC\_Type, [40](#)
  - RTC\_TypeDef, [48](#)
  - TIM\_TypeDef, [56](#)
  - USART\_TypeDef, [67](#)
- RESERVED1
  - RTC\_TypeDef, [48](#)
  - TIM\_TypeDef, [56](#)
  - USART\_TypeDef, [67](#)
- RESERVED10
  - TIM\_TypeDef, [56](#)
- RESERVED11
  - TIM\_TypeDef, [56](#)
- RESERVED12
  - TIM\_TypeDef, [56](#)
- RESERVED13
  - TIM\_TypeDef, [56](#)
- RESERVED14
  - TIM\_TypeDef, [56](#)
- RESERVED15
  - TIM\_TypeDef, [57](#)
- RESERVED16
  - TIM\_TypeDef, [57](#)
- RESERVED17
  - TIM\_TypeDef, [57](#)
- RESERVED18
  - TIM\_TypeDef, [57](#)
- RESERVED19
  - TIM\_TypeDef, [57](#)
- RESERVED2
  - NVIC\_Type, [40](#)
  - RTC\_TypeDef, [48](#)
  - TIM\_TypeDef, [57](#)
  - USART\_TypeDef, [67](#)
- RESERVED3
  - NVIC\_Type, [40](#)
  - RTC\_TypeDef, [49](#)
  - TIM\_TypeDef, [57](#)
  - USART\_TypeDef, [67](#)
- RESERVED4
  - NVIC\_Type, [40](#)
  - RTC\_TypeDef, [49](#)
  - TIM\_TypeDef, [57](#)
  - USART\_TypeDef, [67](#)
- RESERVED5
  - NVIC\_Type, [40](#)
  - RTC\_TypeDef, [49](#)
  - TIM\_TypeDef, [58](#)
  - USART\_TypeDef, [67](#)
- RESERVED6
  - RTC\_TypeDef, [49](#)
  - TIM\_TypeDef, [58](#)
  - USART\_TypeDef, [68](#)
- RESERVED7
  - RTC\_TypeDef, [49](#)
  - TIM\_TypeDef, [58](#)
- RESERVED8
  - RTC\_TypeDef, [49](#)
  - TIM\_TypeDef, [58](#)
- RESERVED9
  - RTC\_TypeDef, [49](#)
  - TIM\_TypeDef, [58](#)
- reset
  - Controller, [16](#)
  - Encoder, [20](#)
  - Kinematic, [30](#)
  - Motor, [37](#)
- Robot
  - Controller, [17](#)
- robotRun
  - SERIAL\_COMMUNICATION.h, [105](#)
- robotStop

- SERIAL\_COMMUNICATION.h, 105
- RPM\_value
  - malha\_fechada.cpp, 86
- rpmToRads
  - MOTOR.h, 102
- rpmToV
  - MOTOR.h, 102
- RSERVED1
  - NVIC\_Type, 41
- RTC
  - Micro.h, 94
- RTC\_BASE
  - Micro.h, 95
- RTC\_IRQn
  - Micro.h, 99
- RTC\_TypeDef, 46
  - ALRH, 47
  - ALRL, 47
  - CNTH, 47
  - CNTL, 47
  - CRH, 47
  - CRL, 48
  - DIVH, 48
  - DIVL, 48
  - PRLH, 48
  - PRL, 48
  - RESERVED0, 48
  - RESERVED1, 48
  - RESERVED2, 48
  - RESERVED3, 49
  - RESERVED4, 49
  - RESERVED5, 49
  - RESERVED6, 49
  - RESERVED7, 49
  - RESERVED8, 49
  - RESERVED9, 49
- RTCAlarm\_IRQn
  - Micro.h, 100
- samplingTime
  - SERIAL\_COMMUNICATION.h, 106
- SCB
  - Micro.h, 95
- SCB\_BASE
  - Micro.h, 95
- SCnSCB
  - Micro.h, 95
- SCS\_BASE
  - Micro.h, 95
- Send
  - USART, 65
- Send\_Vec\_16
  - USART, 65
- sendEverything
  - SERIAL\_COMMUNICATION.h, 106
- sendFloat
  - USART, 65
- sendingTime
  - SERIAL\_COMMUNICATION.h, 106
- sendLineReading
  - SERIAL\_COMMUNICATION.h, 106
- sendMotorDdata
  - SERIAL\_COMMUNICATION.h, 106
- sendMotorsSpeed
  - SERIAL\_COMMUNICATION.h, 106
- sendMotorsSpeedAndControl
  - SERIAL\_COMMUNICATION.h, 106
- sendMotorsSpeedControlAndErrors
  - SERIAL\_COMMUNICATION.h, 106
- sendPositionAndOrientation
  - SERIAL\_COMMUNICATION.h, 107
- sendPWMandEncoderDataRight
  - SERIAL\_COMMUNICATION.h, 107
- sendUInt16
  - USART, 65
- sendUInt32
  - USART, 65
  - USART.cpp, 111
- sendV\_W
  - SERIAL\_COMMUNICATION.h, 107
- sendV\_W\_and\_V\_Wcontrol
  - SERIAL\_COMMUNICATION.h, 107
- sendV\_Wcontrol
  - SERIAL\_COMMUNICATION.h, 107
- sensorFilterOrder
  - LINE\_SENSOR.h, 82
- Sensors
  - Line\_Sensor, 34
- SERIAL\_COMMUNICATION.h
  - bitsToSend, 105
  - robotRun, 105
  - robotStop, 105
  - samplingTime, 106
  - sendEverything, 106
  - sendingTime, 106
  - sendLineReading, 106
  - sendMotorDdata, 106
  - sendMotorsSpeed, 106
  - sendMotorsSpeedAndControl, 106
  - sendMotorsSpeedControlAndErrors, 106
  - sendPositionAndOrientation, 107
  - sendPWMandEncoderDataRight, 107
  - sendV\_W, 107
  - sendV\_W\_and\_V\_Wcontrol, 107
  - sendV\_Wcontrol, 107
- Set\_Speed
  - Motor, 37
- SetADCChannel
  - ADC, 8
- SetGPIOMode
  - GPIO, 25
- SetGPIOPortPin
  - GPIO, 25
- SetIOPin
  - GPIO, 25
- setSpeed
  - Kinematic, 30



- setSpeedRef
  - Controller, [16](#)
- SetTim
  - Timer, [61](#)
- SetTIMChannel
  - Timer, [62](#)
- SetTIMMode
  - Timer, [62](#)
- SetTIMRemap
  - Timer, [62](#)
- SINGLE\_CONVERSION
  - ADC.h, [70](#)
- SMCR
  - TIM\_TypeDef, [58](#)
- SMPR1
  - ADC\_TypeDef, [11](#)
- SMPR2
  - ADC\_TypeDef, [11](#)
- Speed\_Reference
  - Motor, [38](#)
- SPI1\_IRQn
  - Micro.h, [100](#)
- SPI2\_IRQn
  - Micro.h, [100](#)
- SQR1
  - ADC\_TypeDef, [11](#)
- SQR2
  - ADC\_TypeDef, [11](#)
- SQR3
  - ADC\_TypeDef, [11](#)
- SR
  - ADC\_TypeDef, [11](#)
  - FLASH\_TypeDef, [21](#)
  - TIM\_TypeDef, [58](#)
  - USART\_TypeDef, [68](#)
- SRAM\_BASE
  - Micro.h, [95](#)
- SRAM\_BB\_BASE
  - Micro.h, [95](#)
- start
  - Controller, [16](#)
- steps
  - malha\_fechada.cpp, [84](#)
- stepTime
  - malha\_fechada.cpp, [84](#)
- STIR
  - NVIC\_Type, [41](#)
- stop
  - Controller, [17](#)
- SVCall\_IRQn
  - Micro.h, [98](#)
- SysClock, [50](#)
  - MCO, [50](#)
  - SysClock, [50](#)
  - SysClockInit, [50](#)
  - SysTickGetEvent, [51](#)
  - SysTickInit, [51](#)
- SysClock.h
  - BASE\_100ms, [108](#)
  - BASE\_100us, [108](#)
  - BASE\_10ms, [108](#)
  - BASE\_10us, [108](#)
  - BASE\_1ms, [108](#)
  - BASE\_1us, [108](#)
  - SysTickBaseTimeEnum, [108](#)
- SysClockInit
  - SysClock, [50](#)
- SysTick
  - Micro.h, [95](#)
- SysTick\_BASE
  - Micro.h, [96](#)
- SysTick\_IRQn
  - Micro.h, [98](#)
- SysTick\_Type, [52](#)
  - CALIB, [52](#)
  - CTRL, [52](#)
  - LOAD, [52](#)
  - VAL, [52](#)
- SysTickBaseTimeEnum
  - SysClock.h, [108](#)
- SysTickGetEvent
  - SysClock, [51](#)
- SysTickInit
  - SysClock, [51](#)
- t\_ini
  - malha\_fechada.cpp, [86](#)
- TAMPER\_IRQn
  - Micro.h, [99](#)
- testTime
  - malha\_fechada.cpp, [84](#)
- Ticks\_till\_int
  - ENCODER.h, [72](#)
- ticksToRad
  - ENCODER.h, [72](#)
- TIM
  - Timer, [64](#)
- TIM1
  - Micro.h, [96](#)
- TIM1\_BASE
  - Micro.h, [96](#)
- TIM1\_BRK\_IRQn
  - Micro.h, [99](#)
- TIM1\_CC\_IRQn
  - Micro.h, [99](#)
- TIM1\_TRG\_COM\_IRQn
  - Micro.h, [99](#)
- TIM1\_UP\_IRQHandler
  - malha\_fechada.cpp, [85](#)
- TIM1\_UP\_IRQn
  - Micro.h, [99](#)
- TIM2
  - Micro.h, [96](#)
- TIM2\_BASE
  - Micro.h, [96](#)
- TIM2\_IRQHandler
  - malha\_fechada.cpp, [85](#)

- TIM2\_IRQn
  - Micro.h, [99](#)
- TIM3
  - Micro.h, [96](#)
- TIM3\_BASE
  - Micro.h, [96](#)
- TIM3\_IRQHandler
  - malha\_fechada.cpp, [85](#)
- TIM3\_IRQn
  - Micro.h, [100](#)
- TIM4
  - Micro.h, [96](#)
- TIM4\_BASE
  - Micro.h, [97](#)
- TIM4\_IRQHandler
  - malha\_fechada.cpp, [85](#)
- TIM4\_IRQn
  - Micro.h, [100](#)
- TIM\_CH1
  - TIMER.h, [110](#)
- TIM\_CH2
  - TIMER.h, [110](#)
- TIM\_CH3
  - TIMER.h, [110](#)
- TIM\_CH4
  - TIMER.h, [110](#)
- TIM\_CHANNELS
  - TIMER.h, [110](#)
- TIM\_MODE
  - TIMER.h, [110](#)
- TIM\_REMAP
  - TIMER.h, [110](#)
- TIM\_TypeDef, [53](#)
  - ARR, [53](#)
  - BDTR, [54](#)
  - CCER, [54](#)
  - CCMR1, [54](#)
  - CCMR2, [54](#)
  - CCR1, [54](#)
  - CCR2, [54](#)
  - CCR3, [54](#)
  - CCR4, [54](#)
  - CNT, [55](#)
  - CR1, [55](#)
  - CR2, [55](#)
  - DCR, [55](#)
  - DIER, [55](#)
  - DMAR, [55](#)
  - EGR, [55](#)
  - PSC, [55](#)
  - RCR, [56](#)
  - RESERVED0, [56](#)
  - RESERVED1, [56](#)
  - RESERVED10, [56](#)
  - RESERVED11, [56](#)
  - RESERVED12, [56](#)
  - RESERVED13, [56](#)
  - RESERVED14, [56](#)
  - RESERVED15, [57](#)
  - RESERVED16, [57](#)
  - RESERVED17, [57](#)
  - RESERVED18, [57](#)
  - RESERVED19, [57](#)
  - RESERVED2, [57](#)
  - RESERVED3, [57](#)
  - RESERVED4, [57](#)
  - RESERVED5, [58](#)
  - RESERVED6, [58](#)
  - RESERVED7, [58](#)
  - RESERVED8, [58](#)
  - RESERVED9, [58](#)
  - SMCR, [58](#)
  - SR, [58](#)
- time
  - malha\_fechada.cpp, [86](#)
- Time\_between\_int
  - TIMER.h, [110](#)
- Time\_between\_int\_milis
  - TIMER.h, [110](#)
- time\_test
  - malha\_fechada.cpp, [86](#)
- Timer, [59](#)
  - ClearInterruptFlag, [60](#)
  - delay, [60](#)
  - GetTim, [60](#)
  - GetTIMChannel, [61](#)
  - GetTime\_milisec, [61](#)
  - GetTime\_usec, [61](#)
  - GetTIMMode, [61](#)
  - GetTIMRemap, [61](#)
  - InterruptTime, [61](#)
  - SetTim, [61](#)
  - SetTIMChannel, [62](#)
  - SetTIMMode, [62](#)
  - SetTIMRemap, [62](#)
  - TIM, [64](#)
  - Timer, [60](#)
  - Timer\_Handler, [62](#)
  - Timer\_Initiallize, [62](#)
  - TimerInit, [62](#)
  - verifyTimeInterrupt, [63](#)
- TIMER.h
  - AutoReload\_Counter, [109](#)
  - COUNTER, [110](#)
  - FULL\_REMAP, [111](#)
  - NO\_REMAP, [111](#)
  - PARTIAL\_REMAP1, [111](#)
  - PARTIAL\_REMAP2, [111](#)
  - Prescale, [109](#)
  - PWM\_MODE, [110](#)
  - QUADRATURE\_ENCODER\_MODE, [110](#)
  - TIM\_CH1, [110](#)
  - TIM\_CH2, [110](#)
  - TIM\_CH3, [110](#)
  - TIM\_CH4, [110](#)
  - TIM\_CHANNELS, [110](#)

- TIM\_MODE, [110](#)
- TIM\_REMAP, [110](#)
- Time\_between\_int, [110](#)
- Time\_between\_int\_milis, [110](#)
- Timer\_Frequency
  - PWM.h, [104](#)
- Timer\_Handler
  - Timer, [62](#)
- Timer\_Initialize
  - Timer, [62](#)
- TimerInit
  - Timer, [62](#)
- togglePin
  - GPIO, [25](#)
- TPI
  - Micro.h, [97](#)
- TPI\_BASE
  - Micro.h, [97](#)
- Ts
  - MOTOR.h, [102](#)
- ts
  - MOTOR.h, [102](#)
- ts\_w
  - KINEMATIC\_CONTROL.h, [81](#)
- U
  - Motor, [38](#)
- updateLineAngle
  - ANGLE\_LINE\_SENSOR\_H, [13](#)
- updateLineReading
  - Kinematic, [31](#)
- UsageFault\_IRQn
  - Micro.h, [98](#)
- USART, [64](#)
  - Available, [64](#)
  - Receive, [65](#)
  - Send, [65](#)
  - Send\_Vec\_16, [65](#)
  - sendFloat, [65](#)
  - sendUInt16, [65](#)
  - sendUInt32, [65](#)
  - USART, [64](#)
- USART.cpp
  - sendUInt32, [111](#)
- USART.h
  - BD\_1000000, [112](#)
  - BD\_115200, [112](#)
  - BD\_1382400, [112](#)
  - BD\_2250000, [112](#)
  - BD\_230769, [112](#)
  - BD\_250000, [112](#)
  - BD\_38400, [112](#)
  - BD\_57600, [112](#)
  - BD\_9600, [112](#)
  - BD\_ENUM, [112](#)
- USART1
  - Micro.h, [97](#)
- USART1\_BASE
  - Micro.h, [97](#)
- USART1\_IRQn
  - Micro.h, [100](#)
- USART2
  - Micro.h, [97](#)
- USART2\_BASE
  - Micro.h, [97](#)
- USART2\_IRQn
  - Micro.h, [100](#)
- USART3
  - Micro.h, [97](#)
- USART3\_BASE
  - Micro.h, [98](#)
- USART3\_IRQn
  - Micro.h, [100](#)
- USART\_TypeDef, [66](#)
  - BRR, [66](#)
  - CR1, [66](#)
  - CR2, [66](#)
  - CR3, [66](#)
  - DR, [67](#)
  - GTPR, [67](#)
  - RESERVED0, [67](#)
  - RESERVED1, [67](#)
  - RESERVED2, [67](#)
  - RESERVED3, [67](#)
  - RESERVED4, [67](#)
  - RESERVED5, [67](#)
  - RESERVED6, [68](#)
  - SR, [68](#)
- USB\_HP\_CAN1\_TX\_IRQn
  - Micro.h, [99](#)
- USB\_LP\_CAN1\_RX0\_IRQn
  - Micro.h, [99](#)
- USBWakeUp\_IRQn
  - Micro.h, [100](#)
- v\_ref\_ini
  - FOLLOWING\_CONTROLLER.h, [75](#)
- VAL
  - SysTick\_Type, [52](#)
- verifyCommand
  - Communication, [14](#)
- verifyTimeInterrupt
  - Timer, [63](#)
- vtest
  - malha\_fechada.cpp, [86](#)
- w\_ref\_ini
  - FOLLOWING\_CONTROLLER.h, [75](#)
- waitForCommand
  - Communication, [14](#)
- WRPR
  - FLASH\_TypeDef, [21](#)
- wtest
  - malha\_fechada.cpp, [87](#)
- WWDG\_IRQn
  - Micro.h, [98](#)
- xPos

Kinematic, [32](#)  
xtest  
    malha\_fechada.cpp, [87](#)  
yPos  
    Kinematic, [32](#)  
ytest  
    malha\_fechada.cpp, [87](#)