Seguidor

100.1

Generated by Doxygen 1.8.16

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

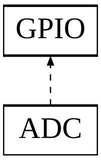
Chapter 4

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
```

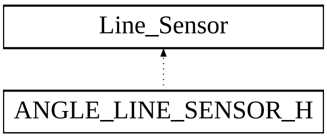
The documentation for this struct was generated from the following file:

- 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_CONTROLER.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
```

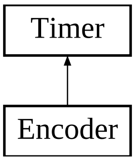
The documentation for this class was generated from the following files:

- C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/FOLLOWING_CONTROLER.h
- C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/FOLLOWING_CONTROLER.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_Initiallize ()
- 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_Initiallize()

```
void Encoder::Encoder_Initiallize ( )  [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
```

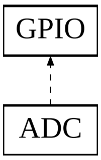
The documentation for this struct was generated from the following file:

- 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 tooglePin ()
- 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

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 tooglePin()

```
void GPIO::tooglePin ( )
```

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}
```

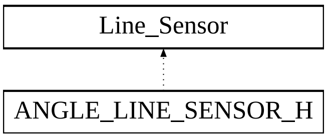
The documentation for this class was generated from the following files:

- 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 & 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 RSERVED1

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

4.15.1.13 STIR

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

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

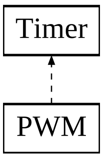
The documentation for this struct was generated from the following file:

- 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 getWritedPWM ()

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 PRLL
- 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 PRLL

```
volatile uint16_t RTC_TypeDef::PRLL
```

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
```

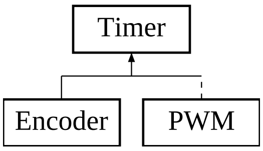
The documentation for this struct was generated from the following file:

- 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 InterrupTime (uint16_t time)
- void ClearInterruptFlag ()

Static Public Member Functions

- static uint32_t GetTime_usec ()
- static uint32_t GetTime_milisec ()
- static void delay (uint32_t delayTime_usec)
- static void Timer_Initiallize ()
- 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 InterrupTime()

```
void Timer::InterrupTime (
             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/Line↩ Follower/ADC.cpp File Reference

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

5.2 C:/Users/bruno/Documents/LineFollower/Project_uVision/Line↩ Follower/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/Line↵Follower/ANGLE_LINE_SENSOR.cpp File Reference

5.4 C:/Users/bruno/Documents/LineFollower/Project_uVision/Line↵Follower/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/Line↵Follower/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 AutoReaload_Ticks 1
- #define Ticks_till_int (AutoReaload_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 AutoReaload_Ticks

```
#define AutoReaload_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 (AutoReaload_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_CONTROLER.cpp File Reference

```
#include "FOLLOWING_CONTROLER.h"
```

5.8 C:/Users/bruno/Documents/LineFollower/Project_uVision/LineFollower/FOLLOWING_CONTROLER.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/Line↩Follower/GPIO.cpp File Reference

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

5.10 C:/Users/bruno/Documents/LineFollower/Project_uVision/Line↩Follower/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_50MH
  = 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/Line↩ Follower/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/Line↩ Follower/main.cpp File Reference

5.17 C:/Users/bruno/Documents/LineFollower/Project_uVision/Line↩ Follower/malha_aberta.cpp File Reference

5.18 C:/Users/bruno/Documents/LineFollower/Project_uVision/Line↩ Follower/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/Line↵Follower/Micro.c File Reference

5.20 C:/Users/bruno/Documents/LineFollower/Project_uVision/Line↵Follower/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
RTCAlarm_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/Line↩Follower/MOTOR.cpp File Reference

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

5.22 C:/Users/bruno/Documents/LineFollower/Project_uVision/Line↩Follower/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/Line↩ Follower/PWM.cpp File Reference

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

5.24 C:/Users/bruno/Documents/LineFollower/Project_uVision/Line↩ Follower/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/Line←Follower/REFLECTANCE_SENSOR.cpp File Reference

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

5.26 C:/Users/bruno/Documents/LineFollower/Project_uVision/Line←Follower/REFLECTANCE_SENSOR.h File Reference

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

Classes

- class Reflectance_Sensor

5.27 C:/Users/bruno/Documents/LineFollower/Project_uVision/Line←Follower/SERIAL_COMMUNICATION.cpp File Reference

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

5.28 C:/Users/bruno/Documents/LineFollower/Project_uVision/Line↩ Follower/SERIAL_COMMUNICATION.h File Reference

```
#include "FOLLOWING_CONTROLER.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/Line↩ Follower/SysClock.cpp File Reference

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

5.30 C:/Users/bruno/Documents/LineFollower/Project_uVision/Line↩ Follower/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

BASE_100ms	
BASE_10ms	
BASE_1ms	
BASE_100us	
BASE_10us	
BASE_1us	

5.31 C:/Users/bruno/Documents/LineFollower/Project_uVision/Line↩ Follower/testes.cpp File Reference

5.32 C:/Users/bruno/Documents/LineFollower/Project_uVision/Line↩ Follower/TIMER.cpp File Reference

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

5.33 C:/Users/bruno/Documents/LineFollower/Project_uVision/Line↩ Follower/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

BD_9600	
BD_38400	
BD_57600	
BD_115200	
BD_230769	
BD_250000	
BD_1382400	
BD_1000000	
BD_2250000	

Index