Bluetooth Controlled R/C Car with Line Follower

MARCH 15

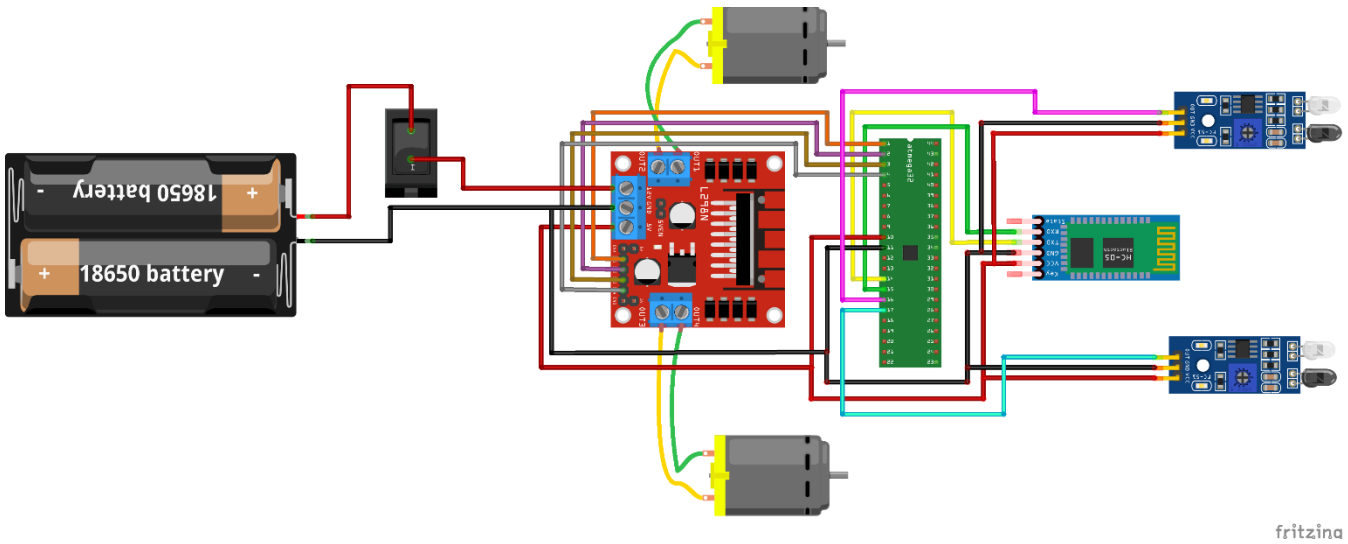
Khaled Ibrahim Abdulaziz

1. Static Design

1.1 Components

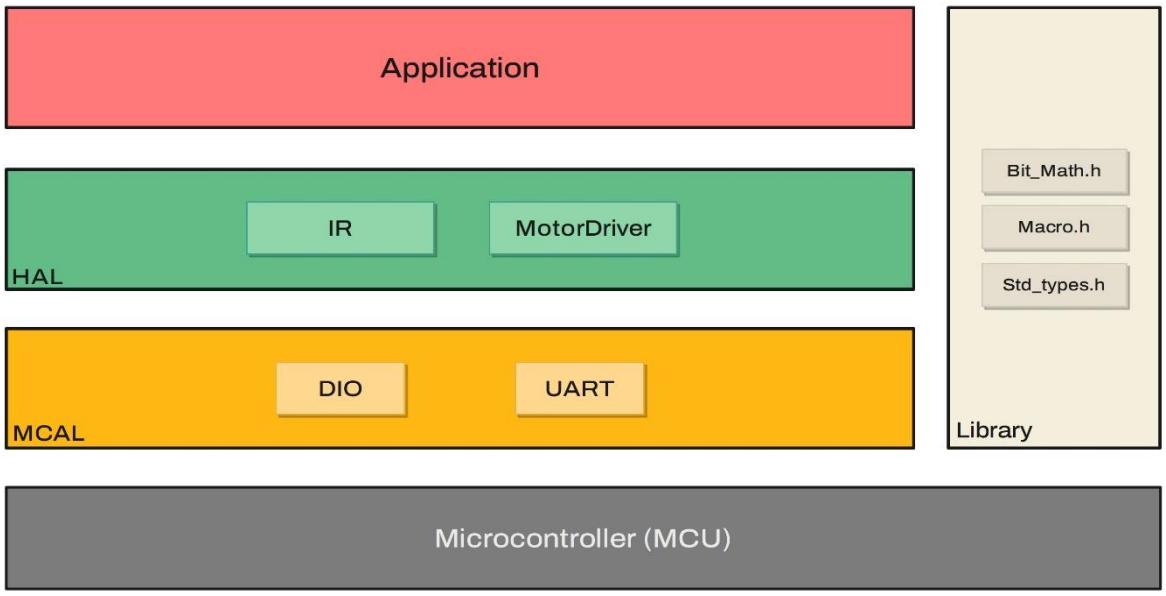
1. 1x AVR Development Board (Atmega32)
2. 1x Motor Driver L298N
3. 2x IR Sensors
4. 1x HC-05 Bluetooth Module
5. 2x DC Motors
6. 1x Car Chassis
7. 2x Li-ion 18650 Battery
8. 1x 2 slots Battery holder
9. 1x 2pin On/Off G130 Boat Rocker Switch

1.2 Schematic



1.3 Layered Architecture

Making the layered architecture and specifying components and modules.



1.2 APIs and Typedefs

1.2. DIO

1.2.1.1 APIs

Name	DIO_u8SetPinDir
Syntax	RequestState DIO_u8SetPinDir(u8 Local_GroupName, u8 Local_PinNumber, u8 Local_PinState);
Description	Set the direction of the PIN
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters(in)	Local_GroupName, Local_PinNumber, Local_PinState
Parameters(out)	None
Return value	RequestState

Name	DIO_u8SetPinValue
Syntax	RequestState DIO_u8SetPinValue(u8 Local_GroupName, u8 Local_PinNumber, u8 Local_PinValue);
Description	Sets the the PIN
Sync/Async	Synchronous
Reentrancy	Non-Reentrant

Parameters(in)	Local_GroupName, Local_PinNumber, Local_PinValue
Parameters(out)	None
Return value	RequestState

Name	DIO_u8GetPinValue
Syntax	RequestState DIO_u8GetPinValue(u8 Local_GroupName, u8 Local_PinNumber, u8 *Local_PinValue);
Description	Reads the value of the PIN
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters(in)	Local_GroupName, Local_PinNumber, Local_PinValue
Parameters(out)	None
Return value	RequestState

Name	DIO_u8SetPortDir
Syntax	RequestState DIO_u8SetPortDir(u8 Local_GroupName, u8 Local_PortState);
Description	Sets the direction of the PORT
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters(in)	Local_GroupName, Local_PortState
Parameters(out)	None
Return value	RequestState

Name	DIO_u8SetPortValue
Syntax	RequestState DIO_u8SetPortValue(u8 Local_GroupName, u8 Local_PortValue);
Description	Sets the PORT with a specific value
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters(in)	Local_GroupName,Local_PortValue
Parameters(out)	None
Return value	RequestState

Name	DIO_u8GetPortValue
Syntax	RequestState DIO_u8GetPortValue(u8 Local_GroupName, u8 *Local_PortValue);
Description	Reads the value of the PORT
Sync/Async	Synchronous

Reentrancy	Non-Reentrant
Parameters(in)	Local_GroupName, Local_PortValue
Parameters(out)	None
Return value	RequestState

Name	DIO_u8ControlPullup
Syntax	RequestState DIO_u8ControlPullup(u8 Local_ConnectionType, u8 Local_GroupName, u8 Local_PinNumber, u8 Local_PullupState);
Description	Enable or Disable PullUp resistor of the PIN
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters(in)	Local_ConnectionType, Local_GroupName, Local_PinNumber, Local_PullupState
Parameters(out)	None
Return value	RequestState

1.2.1.2 Typedefs

Name	RequestState
Type	enum
Description	An enum for handling Errors: RequestHandled, RequestErrorGroupOutOfRange, RequestErrorPinOutOfRange, RequestErrorNotValidState, and RequestErrorNotValidValue

1.2.2 UART

1.2.2.1 APIs

Name	UART_VoidInit
Syntax	void UART_VoidInit(u16 Baudrate);
Description	Initializes UART module with the desired configurations
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters(in)	Baudrate
Parameters(out)	None
Return value	void

Name	UART_TxChar
Syntax	void UART_TxChar(u8 local_data);
Description	Sends a char to the receiver
Sync/Async	Synchronous

Reentrancy	Non-Reentrant
Parameters(in)	local_data
Parameters(out)	None
Return value	void

Name	UART_TxString
Syntax	void UART_TxString(u8 *string);
Description	Sends a string to the receiver
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters(in)	string
Parameters(out)	None
Return value	void

Name	UART_RxChar
Syntax	u8 UART_RxChar(void);
Description	Receives a char from the transmitter
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters(in)	void
Parameters(out)	None
Return value	u8

Name	UART_RxString
Syntax	void UART_RxString(char* string);
Description	Receives a string from the transmitter
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters(in)	string
Parameters(out)	None
Return value	void

1.2.3 MotorDriver

1.2.3.1 APIs

Name	MDriver_init
Syntax	void MDriver_init(void);
Description	Initializes Motor Driver module
Sync/Async	Synchronous
Reentrancy	Non-Reentrant

Parameters(in)	void
Parameters(out)	None
Return value	void

Name	MDriver_MoveBackward
Syntax	void MDriver_MoveBackward(void);
Description	Moves the car backward
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters(in)	void
Parameters(out)	None
Return value	void

Name	MDriver_MoveRight
Syntax	void MDriver_MoveRight(void);
Description	Moves the car to the right
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters(in)	void
Parameters(out)	None
Return value	void

Name	MDriver_MoveLeft
Syntax	void MDriver_MoveLeft(void);
Description	Moves the car to the left
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters(in)	void
Parameters(out)	None
Return value	void

Name	MDriver_MoveForRight
Syntax	void MDriver_MoveForRight(void);
Description	Moves the car forward right
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters(in)	void
Parameters(out)	None
Return value	void

Name	MDriver_MoveForLeft
Syntax	void MDriver_MoveForLeft(void);
Description	Moves the car forward left
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters(in)	void
Parameters(out)	None
Return value	void

Name	MDriver_MoveBackRight
Syntax	void MDriver_MoveBackRight(void);
Description	Moves the car backward right
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters(in)	void
Parameters(out)	None
Return value	void

Name	MDriver_MoveBackLeft
Syntax	void MDriver_MoveBackLeft(void);
Description	Moves the car backward left
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters(in)	void
Parameters(out)	None
Return value	void

Name	MDriver_Stop
Syntax	void MDriver_Stop(void);
Description	Stops the car
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters(in)	void
Parameters(out)	None
Return value	void

1.2.4 IR

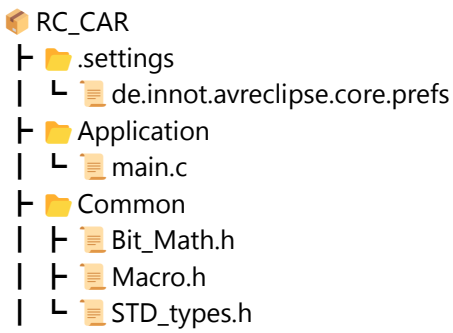
1.2.4.1 APIs

Name	IRSensors_init
Syntax	void IRSensors_init(void);
Description	Initializes IR Sensors
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters(in)	void
Parameters(out)	None
Return value	void

Name	Read_IRSensor_Right
Syntax	u8 Read_IRSensor_Right(void);
Description	Reads the value of the Right IR Sensor
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters(in)	void
Parameters(out)	None
Return value	u8

Name	Read_IRSensor_Left
Syntax	u8 Read_IRSensor_Left(void);
Description	Reads the value of the Left IR Sensor
Sync/Async	Synchronous
Reentrancy	Non-Reentrant
Parameters(in)	void
Parameters(out)	None
Return value	u8

1.3 Folder Structure



```
├ Debug
|  ├ Application
|  |  ├ main.d
|  |  ├ main.o
|  |  └ subdir.mk
|  ├ HAL
|  |  ├ IR
|  |  |  └ Source
|  |  |  |  ├ IR_Program.d
|  |  |  |  ├ IR_Program.o
|  |  |  |  └ subdir.mk
|  |  └ MotorDriver
|  |  |  └ Source
|  |  |  |  ├ MotorDriver_Program.d
|  |  |  |  ├ MotorDriver_Program.o
|  |  |  |  └ subdir.mk
|  ├ MCAL
|  |  ├ DIO
|  |  |  └ Source
|  |  |  |  ├ DIO_Imp.d
|  |  |  |  ├ DIO_Imp.o
|  |  |  |  └ subdir.mk
|  |  └ UART
|  |  |  └ Source
|  |  |  |  ├ subdir.mk
|  |  |  |  ├ UART_program.d
|  |  |  |  └ UART_program.o
|  ├ DriversRepo.lss
|  ├ DriversRepo.map
|  ├ makefile
|  ├ objects.mk
|  ├ RC_CAR.elf
|  ├ RC_CAR.hex
|  ├ RC_CAR.lss
|  ├ RC_CAR.map
|  └ sources.mk
├ HAL
|  ├ IR
|  |  ├ Header
|  |  |  ├ IR_Config.h
|  |  |  └ IR_Interface.h
|  |  └ Source
|  |  |  └ IR_Program.c
|  └ MotorDriver
|  |  ├ Header
|  |  |  ├ MotorDriver_Config.h
|  |  |  └ MotorDriver_Interface.h
|  |  └ Source
|  |  |  └ MotorDriver_Program.c
├ MCAL
|  ├ DIO
|  |  ├ Header
```

```
|  |  |  ├ DIO_Config.h
|  |  |  ├ DIO_Interface.h
|  |  |  ├ DIO_Private.h
|  |  |  └ DIO_Registers.h
|  |  └ Source
|  |  |  └ DIO_Imp.c
|  └ UART
|  |  ├ Header
|  |  |  ├ UART_Config.h
|  |  |  ├ UART_Interface.h
|  |  |  ├ UART_Private.h
|  |  |  └ UART_Register.h
|  |  └ Source
|  |  |  └ UART_program.c
├ .cproject
└ .project
```