



# Static Analysis for UART Driver

PRESENTED BY: AHMED HASSAN

## Agenda:

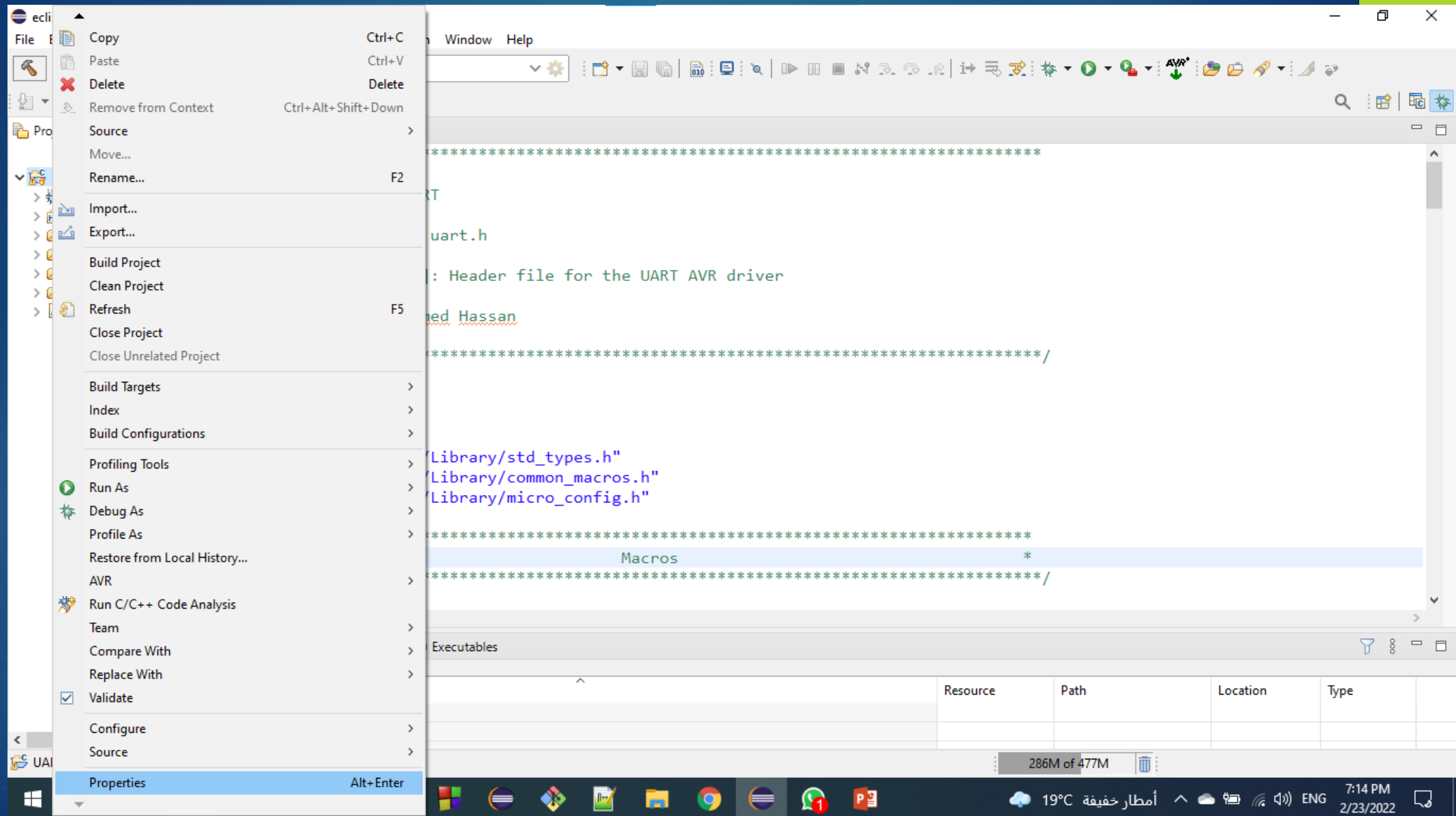
- IDE and Install Static Analysis Tool.
- Warnings and Issues.
- Solution.

## ■ IDE and Install Static Analysis Tool:

Steps of adding static Analysis to IDE:

- ▶ IDE: Eclipse
- ▶ Add the Static Analysis tool to the project:
  - ▶ Step 1: Go to the project property.
  - ▶ Step 2: Select code analysis, then select use project setting.
  - ▶ Step 3: Select the check box that compatible with Misra C Rules.

## Step 1:



## Step 2:

Properties for UART

type filter text

- > Resource
- > AVR
- Builders
- > C/C++ Build
- > C/C++ General
  - Code Analysis**
  - Documentation
  - File Types
  - Formatter
  - Indexer
  - Language Mappings
  - Paths and Symbols
  - Preprocessor Include Pat
  - Profiling Categories
- Git
- Linux Tools Path
- Project Natures
- Project References
- Run/Debug Settings
- Task Repository
- Task Tags
- > Validation
- WikiText

**Code Analysis**

☐ Use workspace settings

☒ Use project settings

[Configure Workspace Settings...](#)

Problems

type filter text

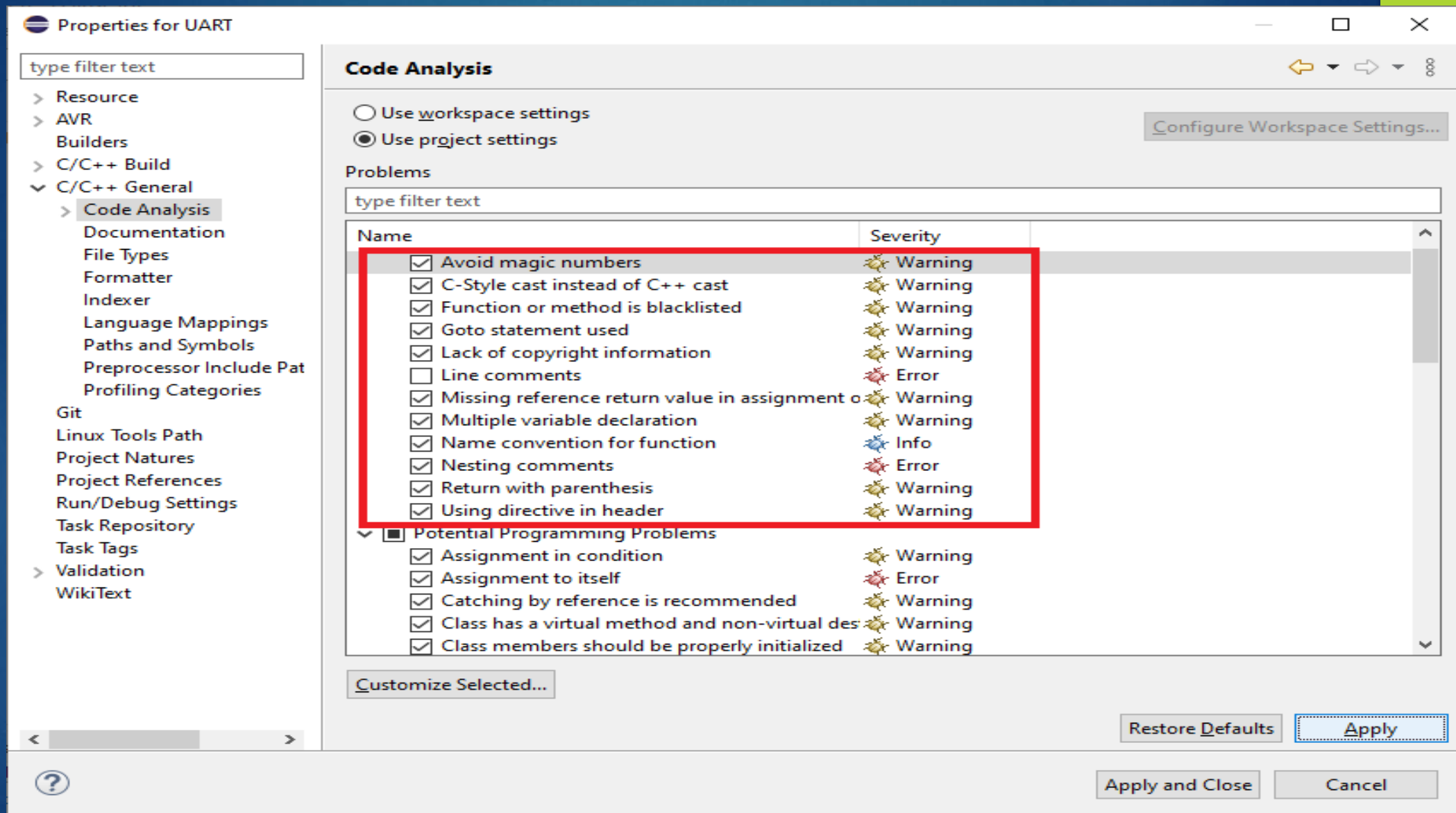
Name	Severity
<input type="checkbox"/> Avoid magic numbers	Warning
<input type="checkbox"/> C-Style cast instead of C++ cast	Warning
<input type="checkbox"/> Function or method is blacklisted	Warning
<input type="checkbox"/> Goto statement used	Warning
<input type="checkbox"/> Lack of copyright information	Warning
<input type="checkbox"/> Line comments	Error
<input type="checkbox"/> Missing reference return value in assignment o	Warning
<input type="checkbox"/> Multiple variable declaration	Warning
<input type="checkbox"/> Name convention for function	Info
<input type="checkbox"/> Nesting comments	Error
<input type="checkbox"/> Return with parenthesis	Warning
<input type="checkbox"/> Using directive in header	Warning
▼ <input checked="" type="checkbox"/> Potential Programming Problems	
<input checked="" type="checkbox"/> Assignment in condition	Warning
<input checked="" type="checkbox"/> Assignment to itself	Error
<input checked="" type="checkbox"/> Catching by reference is recommended	Warning
<input checked="" type="checkbox"/> Class has a virtual method and non-virtual des	Warning
<input checked="" type="checkbox"/> Class members should be properly initialized	Warning

[Customize Selected...](#)

[Restore Defaults](#) [Apply](#)

[Apply and Close](#) [Cancel](#)

## Step 3:





## ■ Warnings and Issues:

<div> <div>Console</div> <div>Problems</div> <div>Executables</div> </div> <div>0 errors, 23 warnings, 10 others</div>				
Description	Resource	Path	Location	Type
<div> <div>⌵</div> <div>⚠ Warnings (23 items)</div> </div>				
⚠ #warning "Compiler optimizations disabled; functions from <util/delay.h> won't work as designed"	UART		line 90, external ...	C/C++ Probl...
⚠ Avoid constant literals	App.c	/UART/Application	line 19	Code Analysi...
⚠ Avoid constant literals	uart.c	/UART/MCAL/UART	line 43	Code Analysi...
⚠ Avoid constant literals	uart.c	/UART/MCAL/UART	line 49	Code Analysi...
⚠ Avoid constant literals	uart.c	/UART/MCAL/UART	line 49	Code Analysi...
⚠ Avoid constant literals	uart.c	/UART/MCAL/UART	line 68	Code Analysi...
⚠ Avoid constant literals	uart.c	/UART/MCAL/UART	line 68	Code Analysi...
⚠ Avoid constant literals	uart.c	/UART/MCAL/UART	line 71	Code Analysi...
⚠ Avoid constant literals	uart.c	/UART/MCAL/UART	line 71	Code Analysi...
⚠ Avoid constant literals	uart.c	/UART/MCAL/UART	line 74	Code Analysi...
⚠ Avoid constant literals	uart.c	/UART/MCAL/UART	line 74	Code Analysi...
⚠ Avoid constant literals	uart.c	/UART/MCAL/UART	line 83	Code Analysi...
⚠ Avoid constant literals	uart.c	/UART/MCAL/UART	line 83	Code Analysi...
⚠ Avoid constant literals	uart.c	/UART/MCAL/UART	line 84	Code Analysi...
⚠ Avoid constant literals	uart.c	/UART/MCAL/UART	line 87	Code Analysi...
⚠ Avoid constant literals	uart.c	/UART/MCAL/UART	line 87	Code Analysi...
⚠ Avoid constant literals	uart.c	/UART/MCAL/UART	line 88	Code Analysi...
⚠ Avoid constant literals	uart.c	/UART/MCAL/UART	line 227	Code Analysi...
⚠ Avoid constant literals	uart.c	/UART/MCAL/UART	line 227	Code Analysi...
⚠ Avoid constant literals	uart.c	/UART/MCAL/UART	line 248	Code Analysi...
⚠ Avoid constant literals	uart.c	/UART/MCAL/UART	line 248	Code Analysi...
⚠ Avoid constant literals	uart.c	/UART/MCAL/UART	line 269	Code Analysi...
⚠ Avoid constant literals	uart.c	/UART/MCAL/UART	line 271	Code Analysi...
<div> <div>&gt;</div> <div>i Infos (10 items)</div> </div>				

## ■ Warnings and Issues:

Console Problems Executables				
0 errors, 23 warnings, 10 others				
Description	Resource	Path	Location	Type
> ⚠ Warnings (23 items)				
v i Infos (10 items)				
💡 Bad function name "App_init" (pattern /^[a-z]/)	App.c	/UART/Application	line 40	Code Analyysi...
💡 Bad function name "App_Update" (pattern /^[a-z]/)	App.c	/UART/Application	line 61	Code Analyysi...
💡 Bad function name "UART_Delnit" (pattern /^[a-z]/)	uart.c	/UART/MCAL/UART	line 266	Code Analyysi...
💡 Bad function name "UART_init" (pattern /^[a-z]/)	uart.c	/UART/MCAL/UART	line 40	Code Analyysi...
💡 Bad function name "UART_receiveString" (pattern /^[a-z]/)	uart.c	/UART/MCAL/UART	line 196	Code Analyysi...
💡 Bad function name "UART_recieveByte" (pattern /^[a-z]/)	uart.c	/UART/MCAL/UART	line 138	Code Analyysi...
💡 Bad function name "UART_sendByte" (pattern /^[a-z]/)	uart.c	/UART/MCAL/UART	line 108	Code Analyysi...
💡 Bad function name "UART_sendString" (pattern /^[a-z]/)	uart.c	/UART/MCAL/UART	line 169	Code Analyysi...
💡 Bad function name "UART_setParityType" (pattern /^[a-z]/)	uart.c	/UART/MCAL/UART	line 224	Code Analyysi...
💡 Bad function name "UART_setStopBit" (pattern /^[a-z]/)	uart.c	/UART/MCAL/UART	line 245	Code Analyysi...



## ■ Solution:

Issue: `void UART_init(const UART_ConfigType * Config_Ptr)`

Solution: `void uart_init(const UART_ConfigType * Config_Ptr)`

Issue: `void UART_sendByte(const uint8_t au8_data)`

Solution: `void uart_sendByte(const uint8_t au8_data)`

Issue: `uint8_t UART_recieveByte(void)`

Solution: `uint8_t uart_recieveByte(void)`

Issue: `void UART_sendString(const uint8_t * aStr_message)`

Solution: `void uart_sendString(const uint8_t * aStr_message)`

Issue: `void UART_receiveString(uint8_t * aStr_message)`

Solution: `void uart_receiveString(uint8_t * aStr_message)`

## ■ Solution:

Issue: `void UART_setParityType(const enuUART_ParityType_t enu_parityType)`

Solution: `void uart_setParityType(const enuUART_ParityType_t enu_parityType)`

Issue: `void UART_setStopBit(const enuUART_StopBit_t enu_stopBit)`

Solution: `void uart_setStopBit(const enuUART_StopBit_t enu_stopBit)`

Issue: `void UART_DeInit(void)`

Solution: `void uart_DeInit(void)`

Issue: 

```
void uart_init(const UART_ConfigType * Config_Ptr)
{
    /* Insert the required Mode in U2X bit in UCSRA Register */
    UCSRA = (UCSRA & 0xFD) | ( (Config_Ptr->uart_Mode) << 1);
```

Solution: 

```
void uart_init(const UART_ConfigType * Config_Ptr)
{
    /* Insert the required Mode in U2X bit in UCSRA Register */
    UCSRA = (UCSRA & MODE_MASK) | ( (Config_Ptr->uart_Mode) << 1);
```

## ■ Solution:

Issue:

```
/* Insert the last bit of required data bits in bit UCSZ2 in UCSRB Register */
UCSRB = (UCSRB & 0xFB) | ( (Config_Ptr->uart_DataBits) & 0x04);
```

Solution:

```
/* Insert the last bit of required data bits in bit UCSZ2 in UCSRB Register */
UCSRB = (UCSRB & NINE_DATA_BITS_MASK) | ( (Config_Ptr->uart_DataBits) & NINE_DATA_BITS_SHIFT);
```

Issue:

```
/*
 * Insert the First two bits of required data bits in bits (UCSZ0, UCSZ1)
 * in UCSRC Register in case of (5,6,7 or 8) Data bits
 */
UCSRC = (UCSRC & 0xF9) | ( (Config_Ptr->uart_DataBits & 0x03) << 1);

/* Insert the required parity type in bits (UPM0, UPM1) in UCSRC Register */
UCSRC = (UCSRC & 0xCF) | ( (Config_Ptr->uart_Parity) << 4UL);

/* Insert the required number of stop bit(s) in bit USBS in UCSRC Register */
UCSRC = (UCSRC & 0xF7) | ( (Config_Ptr->uart_StopBit) << 3UL);
```

Solution:

```
/*
 * Insert the First two bits of required data bits in bits (UCSZ0, UCSZ1)
 * in UCSRC Register in case of (5,6,7 or 8) Data bits
 */
UCSRC = (UCSRC & DATA_BITS_MASK) | ( (Config_Ptr->uart_DataBits & DATA_BITS_SHIFT) << 1);

/* Insert the required parity type in bits (UPM0, UPM1) in UCSRC Register */
UCSRC = (UCSRC & PARITY_MASK) | ( (Config_Ptr->uart_Parity) << PARITY_SHIFT);

/* Insert the required number of stop bit(s) in bit USBS in UCSRC Register */
UCSRC = (UCSRC & STOP_MASK) | ( (Config_Ptr->uart_StopBit) << STOP_SHIFT);
```

## ■ Solution:

Issue:

```
/*
 * This switch case is to choose between two Modes (Normal, Double Speed),
 * then calculate and insert the required Baud rate in First 8 bits from the
 * BAUD_PRESCALE inside UBRRH and last 4 bits in UBRRL
 */
switch(Config_Ptr -> uart_Mode)
{
case NORMAL_SPEED:  UBRRH = ( (((F_CPU / (Config_Ptr->uart_BaudRate * 16UL))) - 1) >> 8UL);
                    UBRRL = ( (((F_CPU / (Config_Ptr->uart_BaudRate * 16UL))) - 1);
                    break;

case DOUBLE_SPEED:  UBRRH = ( (((F_CPU / (Config_Ptr->uart_BaudRate * 8UL))) - 1) >> 8UL);
                    UBRRL = ( (((F_CPU / (Config_Ptr->uart_BaudRate * 8UL))) - 1);
                    break;
}
```

Solution:

```
/*
 * This switch case is to choose between two Modes (Normal, Double Speed),
 * then calculate and insert the required Baud rate in First 8 bits from the
 * BAUD_PRESCALE inside UBRRH and last 4 bits in UBRRL
 */
switch(Config_Ptr -> uart_Mode)
{
case NORMAL_SPEED:  UBRRH = ( (((F_CPU / (Config_Ptr->uart_BaudRate * NORMAL_SPEED_DIVISOR))) - 1) >> SPEED_SHIFT);
                    UBRRL = ( (((F_CPU / (Config_Ptr->uart_BaudRate * NORMAL_SPEED_DIVISOR))) - 1);
                    break;

case DOUBLE_SPEED:  UBRRH = ( (((F_CPU / (Config_Ptr->uart_BaudRate * DOUBLE_SPEED_DIVISOR))) - 1) >> SPEED_SHIFT);
                    UBRRL = ( (((F_CPU / (Config_Ptr->uart_BaudRate * DOUBLE_SPEED_DIVISOR))) - 1);
                    break;
}
```



## ■ Solution:

Issue:

```
void uart_setParityType(const enuUART_ParityType_t enu_parityType)
{
    /* Insert the required parity type in bits (UPM0, UPM1) in UCSRC Register */
    UCSRC = (UCSRC & PARITY_MASK) | (enu_parityType << PARITY_SHIFT);
}
```

Solution:

```
void uart_setParityType(const enuUART_ParityType_t enu_parityType)
{
    /* Insert the required parity type in bits (UPM0, UPM1) in UCSRC Register */
    UCSRC = (UCSRC & PARITY_MASK) | (enu_parityType << PARITY_SHIFT);
}
```

Issue:

```
void uart_setStopBit(const enuUART_StopBit_t enu_stopBit)
{
    /* Insert the required number of stop bit(s) in bit USBS in UCSRC Register */
    UCSRC = (UCSRC & 0xF7) | (enu_stopBit << 3UL);
}
```

Solution:

```
void uart_setStopBit(const enuUART_StopBit_t enu_stopBit)
{
    /* Insert the required number of stop bit(s) in bit USBS in UCSRC Register */
    UCSRC = (UCSRC & STOP_MASK) | (enu_stopBit << STOP_SHIFT);
}
```



## ■ Solution:

Issue:

```
void uart_DeInit(void)
{
    /* Reset All UART Registers to its initial value */
    UCSRA = 0x20;
    UCSRB = 0;
    UCSRC = 0x86;

    /* Clear UART Baud Rate Registers */
    UBRRH = 0;
    UBRRL = 0;
}
```

Solution:

```
void uart_DeInit(void)
{
    /* Reset All UART Registers to its initial value */
    UCSRA = USCRA_INITIAL_VALUE ;
    UCSRB = 0;
    UCSRC = USCRC_INITIAL_VALUE;

    /* Clear UART Baud Rate Registers */
    UBRRH = 0;
    UBRRL = 0;
}
```



Thank you